A Novel Framework Utilizing Bayesian Networks Structured as Logical Syllogisms to Determine Sufficiency of Early-Stage Ship Design Knowledge Queries

by

Samantha Bunting Taylordean

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Naval Architecture and Marine Engineering)
in the University of Michigan
2021

Doctoral Committee:

Associate Professor David J. Singer, Chair
Associate Professor Matthew D. Collette
Professor Lawrence Seiford
Professor Nickolas Vlahopoulos

Samantha Bunting Taylordean

sbtaylor@umich.edu

ORCID iD: 0000-0002-4390-8890

To my family

ACKNOWLEDGEMENTS

It has been said the cure for everything is salt water: whether it be sweat, tears, or the sea. Certainly all three have gone into this work, and I would be remiss if I were not to thank those who have made this process not only bearable but highly enjoyable.

First and foremost, to my advisor Dr. David Singer: thank you for all your support and guidance over the years. I can not imagine being where I am at now without your mentorship; in fact, I would have never considered a PhD without you pushing me to not settle for mediocrity. I hope this work does justice to our animated research discussions.

Thank you to those friends I found in the NAME department at UM. It certainly helped to have people to turn to for both research and life struggles and successes. Special gratitude to Claire Wincott, James Coller, Sammi Meister, Dr. Hao Yuan, Dr. Mike Sypniewski, and Dr. Conner Goodrum.

Thank you to my Michigan family and friends, who are always supportive and interested in topics that are obscure at the best of times.

Finally, a tremendous thank you to my parents, who encourage me to tackle any challenges head-on, and to Duncan and Andra who are almost as excited for me to graduate as I am! I could not have done this without your loving encouragement every step of the way.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

Modern engineering design is a complex, temporal, path dependent process in which designers generate knowledge for decision-making. Designers do this by utilizing design tools such as optimization or design synthesis tools that generate data to explore a potential design space. There is a misguided equivalence within the naval design community that data, information, and knowledge are interchangeable. It is assumed that design data is a representation of relationships within and between data, information, and knowledge. Knowledge, which is ultimately the goal, is derived from the proper combination of information with analysis and experience. Decision outcomes made based on designer perception of knowledge will not be determined until the impacts of the decision are realized at a later date.

The critical question addressed in this thesis is "how does one determine the sufficiency or quality of a design decision based upon design tool generated data without awareness of future outcomes?" To address this issue, the first realization is that all design tools are created based upon knowledge artifacts. These artifacts and their structure predicate the available data that will be used for design decision-making. Often these tools are modified or extended for use beyond their original development intent. While design data generating tools are assumed to be generalized for a wide range of appropriate use, the ability to determine if this is true does not exist within the literature. One can clearly prove failures in design tool suitability through traditional convergence statistics and lack of Pareto optimality. The issue lies in the regions in which solution generation and solution statistics, as well as Pareto front generation, is deemed successful. However, statistical success does not relate to

data suitability for the desired design knowledge generation and query.

Knowledge queries are uniquely human in that, unlike computers, designers have the ability to follow nonlinear patterns of associations to generate inferences based on data perception. This puts designers at risk for formulating inferences that may not be true relative to the underlying statistics associated with how the inference data was generated. Cognitive biases define the myriad methods in which designers associate meanings to data in ways that do not directly map the data. While the thesis contained within does not directly address cognitive biases it provides a mechanism for the evaluation of design inferences utilizing logical syllogism evaluations of design space exploration data as the means for the determination of sufficiency of generated data used for decision-making.

This dissertation presents the development of a novel logic-based syllogistic Bayesian framework that enables the evaluation of the suitable use of a tool for knowledge development or queries. This work utilizes AAA1, AAA2, and AAA3 syllogisms to represent designer decisions. Syllogisms are transformed into Bayesian networks used to calculate metrics for evaluation of the data generated by an instance of a tool. A unique approach to network construction allows for Bayesian probability and mutual information to examine designer decision-making as syllogisms. Within this work, several cases will be presented that demonstrate the methodology as well as extend the framework to encompass non-standard layered logical structures.

CHAPTER I

Introduction

Modern engineering design is a complex, path dependent process in which knowledge is generated for decision-making through time. Early stage design requires the growth and development of knowledge structures so that decisions can be made that lead to design development progression. Decisions with the greatest impact on the final design are often made in the earliest design stages where there is the least amount of design knowledge and the most uncertainty. Significant time and effort has gone into the development of tools to support the creation of design data used for design knowledge generation, but little has been done to understand the construction and implication of conclusions drawn when using automated tool-generated data.

The research presented in this thesis utilizes Design Science, Bayesian networks, and Logic Theory to create a framework that enables the understanding of the suitability of tool-generated data for use in specific designer inferences whose intent is decision-making. This leads to general strategies to understand how conclusions and thus decisions arise from designer knowledge structures populated in part by the use of data generated by automated design tools. The goal of this framework is to provide new insights into the suitability of design tool-generated data for decision-making queries.

1.1 Motivation and Background

The growth of knowledge for decision-making in design can be best understood as the development of the knowledge structure; that is, the establishment of the relationships between the ideas, concept elements, and evidence behind the product (*Goodrum*, 2020). One piece of building design knowledge is through understanding and formulation of the design problem, often by using tools developed for initial design solution generation and tradespace development. This is expressed in the cyclical growth of knowledge structures and product structures. Product structures describe the constituent connections and interdependent elements which define the product either physically or virtually. At each stage of the design processes, deliverables are established which in turn predicate what knowledge must be created to complete the tasks associated with the deliverable. Once the knowledge structures are formed, decisions are made to populate the product structures from the knowledge structures for beginning the next stage of design.

The interplay between knowledge structures and product structures results in the codification of the path dependencies inherent in knowledge structures (*Page*, 2006). These path dependencies are the results of how the ordering in which the design is approached influences the development of the knowledge structure. Path dependency consequences affect future outcomes of designs in ways such as the shift toward emergent design failures and inadvertent lock-in (*Dong*, 2016). These potentially unappreciated constraints are inherent to requirements elucidation and are integral to the consideration of ship design as a wicked problem (*Andrews*, 2012). Wicked problems were originally coined by *Webber and Rittel* (1973) and summarized by *Conklin* (2006) as six defining characteristics:

1. The problem is not understood until after the formulation of a solution.

2. Design problems have no stopping rule.

3. Solutions to design problems are not right or wrong but good or bad.

4. Every design problem is essentially novel and unique.

5. Every solution to a design problem is a "one shot operation."

6. Design problems have no given alternative solutions.

Each of these characteristics apply to the creation of novel ship designs. While each aspect of the wicked problem is important, it is the belief of the author that the first tenet, 'the problem is not understood until after the formulation of a solution', is the most critical when addressing the issues associated with the assumption that knowledge query conclusions derived from design tool output are new, novel, and sufficient. In all problem formulations that are not pure physics representations, the functions or equations and code structure which represents a state of the developer's knowledge structure, is entwined with the product structure represented by the solutions and input-to-output relationships or solution compositions. The actions taken to solve the wicked problem are analogous to those in the pursuit of understanding its nature. When wicked problems are attempted to be understood through the lens of a design tool, the nature of the problem is then defined by the knowledge structure and knowledge structure interrelationships contained within the tool in conjunction with the designer's conclusions drawn through the manipulation of the tool. The question then arises: how is sufficiency of the designer's conclusion quantified in terms of the data generated by a design tool?

Historically the designer's subjectivity and experience is integral to the determination of sufficiency allowing for design decisions and design development progression. For a design decision to be made, the designer must choose some indication by which they judge the solution as 'good enough' or, at a minimum, sufficient given the current state of design knowledge. Common practices include fulfilling a list of requirements or numerical thresholds. These conditions originated as a designer's expert opinion and are used by modern designers in concert with their own acumen to subjectively determine what conditions or circumstances must be reached for a ‘good enough’ solution to end the design activity and move forward in the design process.

Often, designers follow certain standards to mitigate the possibility of emergent design failures. One such potential consequence is inadvertent design lock-in. Lock-in refers to the stage in a design at which it becomes extremely difficult, and thus costly, to substantially change the design path regardless of new information. This is not necessarily bad, depending on when and where it occurs during the process. Should the sequence of actions taken to develop the knowledge structure limit the integration and influence of future events, significant time and money may be expended to return the design to a stage where it can still be altered appreciably.

In order to mitigate undesired consequences from lock-in or other emergent failures, one strategy taken by designers is to avoid unnecessary reductions of the solution set (*Singer et al.*, 2009). By only eliminating options that previous experience or authorities have determined to not contribute to developing the ending solution conditions, the possibility of emergent failure can be lessened.

Thus, solutions selected to move forward with are not always the best or optimal solution. Wicked problems have no “true or false answers ... assessments of proposed solutions are expressed as ‘good’ or ‘bad’ or, more likely, as ‘better or worse’ or ‘satisfying’ or ‘good enough’ ” (*Webber and Rittel*, 1973). A ‘good enough’ solution is one that the designer has determined as acceptable once they have developed a knowledge structure for such decision-making and determined the solution meets the proscribed requirements.

This notion of ‘good enough’ can be described by the concept of sufficiency. A sufficient decision is one that the designer declares adequate based on appropriate knowledge from designer actions and prior experiences. Paradoxically, sufficiency is informed by the problem formulation. This interdependent relationship between problem definition and solution creation strongly influences the construction of the knowledge structure for decision-making. Sufficient decisions are made with knowledge influenced by the accessibility and ordering of information the designer seeks out, but the cyclical nature means that significant back and forth between problem formulation and solution creation occurs until a designer determines enough knowledge has been accumulated for decision-making.

(*Floridi*, 2017) describes this process of determining sufficient conditions as "a past-oriented approach, consistent with causal, genetic, or genealogical forms of reasoning that lead to a particular modelling or conceptualization of reality, with the identification of necessary and (perhaps) jointly sufficient conditions, with investigations about what must have been the case for something else to be the case". These investigations are a facet of designers' attempts to avoid the codification of adverse path dependencies within the solution. Designers must be conscientious of design lock-in or other consequences resulting from the development of knowledge structures.

So how do designers decide when enough knowledge has been accumulated for determining sufficiency? Good decision-making requires a balance between three factors: enough knowledge to advance the design cycle forward without prematurely eliminating potential solutions, avoiding detrimental consequences such as lock in or emergence as a result of path dependencies, and the indiscriminate infinite spend of resources on the design cycle. Thus there is a necessity in seeking out sufficient knowledge rather than complete knowledge in an appropriate way to support decision-making.

This question is key in fields riddled with uncertainty. To determine sufficiency, it is crucial that "decision makers identify whether they have enough information to make a choice" (*Dong and Hayes*, 2011). In complex design, sufficiency is described as a 'you'll know it when you see it' trait. Utilization of expert evaluations is the industry standard of judging when sufficiency has been reached (*Flyvbjerg et al.*, 2009), which does not assist in understanding how exactly an expert determines that.

Additionally, design is path dependent in regards to knowledge structures as historical knowledge can influence future outcomes, both positively and negatively. By not understanding how sufficient decisions come from knowledge, emergent design failures or lock-in can force a design predication before it is suitable, resulting in re-work or other temporal or monetarily expensive repercussions.

To prevent these consequences and assist in the determination of sufficiency, many tools have been developed to aid in the process of knowledge generation. One such tool on which the US Navy has spent significant resources is the Advanced Ship and Submarine Evaluation Tool (ASSET). ASSET is the US Navy's principal concept design synthesis tool used in early stage design to "provide the bulk of the design data and analysis for these designs" (*NAVSEA*, 2012). The US Navy wants to develop tools for design with a focus on tool packages that can generate knowledge for decision-making. "The results of these complex analyses must be visualized and packaged in a way that they are easy to understand by both the design engineers and program managers such that they can be the basis of a smart, timely decision-making process" (*Kassel et al.*, 2010). There is a desire for these tools to produce not just results, but also the understanding behind them in such a way as to mitigate the risk of emergent design failures. This understanding is essential for knowledge structure development and the formation of decisions.

The Whole System Trade Analysis (WSTA) tool was developed and applied by Sandia National Laboratories in partnership with the US Army Program Executive Office for Ground Combat Systems as "a decision support tool". WSTA has "visualizations designed to facilitate rapid and complete understanding of the trade-space to stakeholders and provide drill down capability to supporting rationale" (*Edwards et al.*, 2015) and is advertised as providing "a holistic framework for modeling and understanding [key performance, cost, risk, and growth] tradeoffs." (*Henry et al.*, 2016)

There is an expressed need for "a deeper understanding of how these tools are used and, more importantly, how they can be used ... in support of the decision analysis process" (*Spero et al.*, 2014). It is evident that there is a link between use of these tools in acquiring knowledge for decision-making and desiring suitable tool use, with emphasis on the development of understanding in support of decision-making. *Howard* (1977) sums up the prevailing view: "It becomes increasingly important for a [person] to be able to show people why [they] arrived at a particular decision. It is also important for them to be able to see what changes in factors surrounding that decision might have led to a different decision."

'Changes in factors' are often investigated through use of tools' advertised visualization features such as toggle bars or filtering components. In order to examine appropriateness of suppositions that the designer considers, they may ask 'what-if' questions (eg. 'What if I look at designs with...') to increase knowledge robustness for decision-making. These questions are attempts to examine appropriateness of decisions without committing to a path with potential negative implications and interdependencies.

The overwhelming amount of countless decisions of varying scale to be made at every junction in early stage design magnifies this issue. *Ferguson* (1994) attests that there are "dozens of small decisions and hundreds of tiny ones that every designer makes throughout the whole process of engineering design." The designer is constantly forming conclusions and testing them with every 'what-if' question asked of a tool during the design process. The way that a designer evaluates subjective suitability of her use of a tool for certain sets of decisions is not quantitatively examined when codifying the knowledge structure as a product structure. Attempts to do so are often the application of the knowledge structure to subjectivity rather than the evaluation as seen in the use of utility theory and fuzzy logic.

Utility theory, which originated in economics but has made its way into many fields where subjective value rankings can be valuable, is one way to encode expert opinion or create quantitative metrics for preferences that may lack data or information (*Knight and Singer*, 2014). Publications on optimal decision-making under uncertainty skirt against the issue of subjectivity in decision-making (*Bullock and Logan*, 1969). "The imposition of (subjective) values for resolving conflicts leads to rejecting 'objective' optimality and replacing it with the criterion of consistency with one's goals and values" (*Einhorn and Hogarth*, 1981). Utility theory allows for an improved understanding of design decisions, seen in how options with higher utilities correspond to more expert preference for those options. The knowledge structure is applied as the utility values and weights determined by consulted experts. It lacks any evaluation of the preferences themselves.

A similar absence of an evaluation of preferences is present in fuzzy logic methods which also incorporate subjective perspectives. Fuzzy logic is a generalization of standard logic where instead of designating something as completely True (1) or False (0), it can possess a degree of truth anywhere between zero and one, inclusive. Fuzzy logic can use linguistic variables to incorporate human expertise into engineering models to increase the flexibility of design optimization routines (*Brefort*, 2018), such as in the US Navy Intelligent Ship Arrangement (ISA) tool. "The overall objective of the ISA system development has been to ... assist the arrangements designer to create effective, rationally-based surface ship arrangements with the maximum amount of intelligent decision-making support" (*Parsons et al.*, 2008). The ISA is an optimization tool that uses fuzzy logic to semiautomatically develop arrangements meeting criteria specified by the designer. Fuzzy logic allows for the designer's perspective of uncertainty to be built into a model instead of entirely relying on post-processing reasoning. However, these designer perspectives are not critically examined for appropriateness.

Evaluating designer's subjective decisions remains elusive within tools designed to assist knowledge structure creation. Examination of appropriateness of designer subjective decision-making is instrumental in recognizing which designs are sufficient. Designers want to know the reasoning behind the decisions they make and attempt to use tools to investigate why. The industry invests heavily in tools to increase information to augment knowledge structures for decision-making and mitigate risk but does not similarly invest in understanding appropriateness of the data generated by a tool use for decision-making.

Work in the University of Michigan's Advanced Naval Concepts Research lab approaches this goal. *Shields* (2017) proposes a knowledge-centric perspective of design and uses network analysis to identify a designer's solution development and decision-making behaviors to avoid emergent design failures. *Goodrum* (2020) addresses how knowledge can be modelled and quantified in a design activity to understand conceptual robustness from a knowledge-centric perspective. *Sypniewski* (2019) investigates and uncovers design interdependencies as a product of the knowledge structure of the tools used to create it in order to identify, but not evaluate, suitability.

There is a research gap in evaluating the appropriateness of decision-making derived from tool data. Currently there are limited tools to qualitatively and quantitatively describe and understand decisions made from conclusions and their associated foundational premises. Logic theory can be used to examine and codify conclusions drawn by designers. In combination with design theory, it can lead to general strategies to understand decision-making from conclusions based on knowledge structures and provide new insights into how designers understand the formulation of conclusions to reach sufficiency in tool driven decision-making.

1.2 Research Scope

The presented work is focused on developing a framework to answer the fundamental question: "how can designers determine the sufficiency or quality of a design decision based upon design tool-generated data without awareness of future outcomes?" Evaluating designer decision-making via structures commonplace in logic theory allows for an innovative and insightful understanding of how designers compose and understand decisions.

The primary research questions this dissertation wants to address are as follows:

1. How can design inferences be represented to model and analyze designer decision-making?

2. How can design inferences be quantified and evaluated in the context of designer decision-making?

3. How can sufficiency for data generated by an instance of a tool be understood without awareness of future outcomes?

This work provides a mechanism for the evaluation of design inferences utilizing logical syllogism evaluations of design space exploration data as the means for the determination of sufficiency of generated data used for decision-making. The following section outlines how this work will address these questions.

1.3 Dissertation Structure

The remainder of this dissertation is divided into the following chapters that are organized as follows:

- Chapter II presents background information on logic theory in the context of engineering design.

- Chapter III introduces the logic-based syllogistic Bayesian framework and details the mechanics of syllogism construction, Bayesian methods, and mutual information needed to address the question of sufficient decision-making for standard flat and non-standard layered logical structures.

- Chapter IV presents the case study that demonstrates the methodology's capability to evaluate decision-making sufficiency.

- Chapter V details the contributions of this work and future topics of interest.

CHAPTER II

Logic Theory

Logic theory provides a descriptive way to address the construction of conclusions from knowledge structures for decision-making in early stage design. Logic theory describes the three main types of logical reasoning instances: deduction, induction, and abduction. These three instances work together in a retroductive process to develop a sufficient understanding behind a surprising fact. Retroduction provides a mechanism to describe engineering design and the cyclical nature between knowledge structures and product structures during design stages.

2.1 Deduction

Aristotle and his followers are credited with originating traditional logic theory, which began as deductive reasoning. Deductive reasoning was originally defined as any process of reasoning from one or more premises to reach a conclusion. These contributions of premises to a conclusion are organized as syllogisms. Syllogisms are at the core of historic deductive reasoning, where conclusions are determined by combining existing statements. A classic deductive syllogism example shows this:

(1) if p then q

(2) if q then r

(3) if p then r

If statements (1) and (2) are True, then statement (3) can be derived.

The deductive syllogism is frequently used in mathematical logical proofs. The law of syllogism allows for a mathematician to present a series of statements, each following logically from those previously assumed to be true, to end with what she is trying to prove (*Lazar*, 1938). Deduction has a presupposition of truth as it is used to prove that something *must* be, hence its prevalence in the field of mathematics. It must be noted that deduction cannot lead to the discovery of new knowledge as the conclusion has by definition already been embedded in the premises (*Taylor et al.*, 2018). In addition to mathematics, deductive reasoning can be defined in the context of scientific experimentation as 'top-down' method of applying a known rule or law to some observation or data to produce a result (*Shepherd and Sutcliffe*, 2011). The rule or law is the first statement, the observation or data is the second, and the conclusion is the result of these statements.

This can be illustrated with a trivial example of determining where one goes for dinner. Say you are located in Ann Arbor, Michigan. Your dining companion is a vegetarian. You are already walking down Main Street where you can assume all restaurants have at least one vegetarian dish. These considerations can be laid out in a logically deductive syllogism:

Rule: Main Street restaurants have vegetarian options.

Case: You are at a Main Street restaurant.

Conclusion: This restaurant has a vegetarian option.

Deductive reasoning begins with the acknowledgment of the rule that Main Street restaurants all have vegetarian options. This rule is applied to the specific case that you are at a restaurant on Main Street. This produces the conclusion (hypothesis) that the restaurant you are at has a vegetarian option. From this conclusion, you make the decision that you will eat at this restaurant because it has a vegetarian dish.

Deduction is an essential kind of reasoning, but its process is rigid. What if the vegetarian option available at most Main Street restaurants is only a side salad and not an actual entree? What if you were not sure that all Main Street restaurants had at least one vegetarian dish? Other reasoning types were developed to cover the limitations of deduction. One such type is induction, which can account for situations that incorporate uncertainty.

2.2 Induction

Induction is a 'bottom-up' method of reasoning (*Shepherd and Sutcliffe*, 2011). With induction, the reasoner begins with some data or specific case, generalizing or extrapolating from the case with tests or observations to derive a general rule or law that is epistemically probable.

An instance of applied inductive reasoning can take place as an extension of the previous dinner example. You begin with the same observation as last time: you and your companion are at a Main Street restaurant. You ask and find that the place you are at does not have vegetarian options that are not a side salad after all. You extrapolate from this and infer that there are no suitable vegetarian options in the area. The inductive syllogism associated with this is as follows:

Case: You are at a Main Street restaurant.

Test: This restaurant does not have an appropriate vegetarian option.

Conclusion: It's likely that no other Main Street restaurants have appropriate vegetarian options.

From this conclusion, you make the decision to go someplace not on Main Street for dinner. Despite the different constructions of the syllogisms for the reasoning types of deduction and induction, neither is necessarily superior to the other. An inductive syllogism applies a test to a case in order to make a conclusion while a deductive syllogism implements a case from a rule to draw a conclusion. Different reasoning processes to produce a conclusion result in making different decisions. Syllogisms can codify how premises contribute to a conclusion.

Syllogisms originated from traditional propositional logic which requires that each statement composing the syllogism have a specific truth value of either True or False. These values do not imply correctness of the statements but are related to the logical validity of the syllogism's conclusion. Logical validity is determined only if the argument takes a form which makes it impossible for the premises to be True and the conclusion nevertheless to be False. This is distinct from judging the actual truth value of the premises. Arguments that are logically valid are not required to have premises that are correct, just considered True by the subjective determination of the term definitions.

Logical validity is not an indication of syllogism usefulness for decision-making. Deduction implies a logically valid syllogism while induction does not. Deductive reasoning provides a logically valid process by which if the premises are true the conclusion is also true. The deductive syllogism dinner example is a logically valid construction of a deductive reasoning instance, but does not necessarily imply a correct conclusion. Deductive reasoning based on a wrong premise may lead to a wrong conclusion. Induction, even with correct premises, may lead to a wrong conclusion.

Most inductive reasoning instances cannot be logically valid because even if all premises are known to be True, the conclusion cannot be guaranteed to be True. Induction provides a process by which a likely and probable conclusion can be reached, critical when dealing with uncertainty or the absence of information. Induction yields a conclusion that may not be assumed logically valid but can still be correct. The inductive reasoning example has you conclude that since some Main Street restaurants do not have appropriate vegetarian options, all Main Street restaurants do not. The application of the situation of the restaurant you are at to all restaurants occurs due to the reasoner's perspective. You may consider the probability of the events being related to be high due to previous experience out for dinner in similar circumstances. If you thought that the restaurant you were at was not representative of other Main Street restaurants, you would have drawn a different conclusion than the one presented as an example.

Inductive reasoning can be looked at as the form of pattern recognition based on experience and personal knowledge. This generally requires the application of information that is known or observed (*Taylor et al.*, 2018). The applicability of induction is limited as it can only provide incomplete support for a general rule. The finite number of observations available to test a specific hypothesis (such that the conclusion is being drawn from the support of the the existing knowledge base) means that should there be evidence that would invalidate the inductive conclusion, the conclusion would not have been made. Induction can produce superficial conclusions but cannot get to "the bottom of things," as *Peirce* (1967) says. Why were no appropriate vegetarian options available? What reasoning instance type can be used to find out?

Certain cases lack baseline knowledge or information to draw conclusions from, such as the circumstances for new or novel design problems. Should a conclusion be required outside of an existing schema, inductive reasoning is unable to produce it. These cases require insight or intuition to be formalized and included in a new form of logical reasoning.

2.3 Abduction

Charles Peirce was one of the first logicians to tackle the idea of a formal reasoning instance that codifies intuition. He initially lacked a clear, consistent definition of this type of reasoning but over the next 30 years, Peirce would come to define the reasoning instance that includes intuition as abduction.

Peirce's original description of abduction is as follows: "In the inquiry, all the possible significant circumstances of the surprising phenomenon are mustered and pondered, until a conjecture furnishes some possible Explanation (sic) of it, by which I mean a syllogism exhibiting the surprising fact as necessarily following from the circumstances of its occurrence together with the truth of the conjecture as premises" (*Peirce*, 1967).

This loosely defined formalization of abduction was essentially "inference to the best explanation" (*Sober*, 2013). Abduction typically begins with the appearance of some surprising fact or phenomenon. From these observations, a hypothesis is formed by the reasoner that offers an explanation of the cause behind this phenomenon. Abduction differs from deduction and induction. Deduction assumes certainty, induction deals with probability based on data, but abduction works to provide a best guess approach based on limited information. Generally the abductive reasoning instance produces a conclusion as a suggested statement to test the hypothesis.

Continuing the dinner example, it may turn out that you were only considering restaurants that could seat you immediately. Restaurants that were busy and had wait times were initially ignored by you despite having appropriate vegetarian dinner entrees. You know that Ann Arbor has a large amount of vegetarians. You have a hunch that the reason none of those restaurants were available to eat at was because most groups out to dinner include vegetarians that prefer those restaurants.

Like deduction and induction, abduction can be formalized into a syllogism:

Case: Restaurants available do not have appropriate vegetarian options.

Hypothesis: Vegetarians who go out to dinner eat at places with vegetarian options.

Conclusion: Restaurants with vegetarian options are busy, so the restaurants available do not have those options.

Concluding that vegetarians are why only restaurants without appropriate vegetarian options are available ends the abductive reasoning instance but does not end the thought process. The reasoner needs more tests or observations to evaluate and adjust the hypothesis. Perhaps you decide to interview others who frequent restaurants to see if there is information that supports your hunch. Maybe restaurants that have limited menus (and thus no vegetarian option) have faster table turnover, resulting in shorter wait time. The conclusion of abduction leaves the reasoner in wont of more investigation to test and support the conclusion drawn.

Abduction alone lacks substantiation for the inference it provides. It must be complemented by deduction and induction for substantiation. Abduction can be considered the logic of discovery supported by deduction and induction as the logics of proof (*Hanson*, 1958). In contrast to deduction and induction as application or generalization of natural laws respectively, abductive reasoning instances are "the only ones in which after they have been admitted to be just, it still remains to inquire whether they are advantageous." (*Peirce*, 1967)

Abductive reasoning creates a hypothesis from an existing, accessible base of knowledge. If justification of the hypothesis requires additional data or information to be found, abduction alone cannot be alone to draw conclusions for decision-making.

Abduction seldom provides a logically valid conclusion. Since abduction is used to generate possible explanations and hypotheses for incomplete observations or surprising facts early in the diagnostic process, it acts as a precursor to more rigorous testing of the hypothesis. That testing is provided by induction and deduction. All three of these instances in concert make up the cycle of retroduction.

2.4 Retroduction

The necessity to justify the abductive hypothesis with additional support prompted Peirce to revise and extend abduction into the retroductive cycle. Retroduction is a recursive cycle that includes deduction, induction, and abduction and is composed of three interrelated stages: "finding relevant variables, linking them in a causal chain, and assessing the plausibility of the chain." (*Einhorn and Hogarth*, 1987) Retroduction is a knowledge extending process of drawing logical inferences from data and information as a cycle of thinking backwards from a surprising observation (*Rollier and Turner*, 1994).

Retroduction begins with the apprehension of a surprising fact. From this observation, abduction occurs as the reasoner works backwards to develop a hunch behind the surprise. Using deduction and induction, the hunch is then examined and altered as needed before engendering a hypothesis worthy of rigorous testing. The hypothesis may require additional analysis or adjustment, often needing more testing or further acquisition of information for the development of knowledge. The back and forth between hypothesis articulation and knowledge structure development continues until the concluding solution is deemed sufficient by the reasoner to understand the why of the surprising fact. At that point the knowledge structures are encoded into product structures for future stages of design.

The stopping point of retroduction is the assessment of sufficiency, similar to the way design activities are concluded in engineering design. Historically, engineering design assesses sufficiency from the designer's judgement. *Einhorn and Hogarth* (1981) describes this succinctly as "In the final analysis the outputs of optimal models are evaluated by judgement, i.e. Do we like the outcomes, do we believe the axioms to be reasonable, and should we be content?"

The recursive nature of retroduction is evident in the retrospective process of engineering design which entangles formulation of a solution with understanding of the problem. Therefore this work postulates that engineering design uses a retroductive cycle to generate the knowledge necessary for decision-making. Traditional use of design space tools in early stage design exemplify this. Designers generate data from these tools and use visualization aspects such as toggle bars, lagging indicators, and more to parse the data for drawing inferences necessary for decision-making. When a designer asks 'what-if' questions while searching the design solutions space, she is exploring other circumstances while developing and adjusting hunches for retroductive reasoning to retroactively build knowledge structures for decision-making.

2.5 Retroductive Engineering Design

To better examine engineering design as a retroductive cycle, retroduction can be separated into five steps. These steps loosely follow *Peirce* (1967)'s psuedo-syllogistic construction:

The surprising fact, C, is observed.

But if A were true, C would be a matter of course.

Hence, there is reason to suspect that A is true.

The first and second step of inquiry follow closely along the first premise seen above: the surprising fact, C, is observed. Step one requires some phenomenon is noticed or arises that gives cause to wonder what may be behind it. Step two is the formation of an initial hunch. These steps are generally abductive.

The third step regards the engendering of the hypothesis: if A (the hypothesis) were true, then C (the surprising fact) would follow. The hunch is formalized into a set of statements that compose a testable hypothesis.

The fourth and fifth step deliberate over what the reasons are to suspect that A (the hypothesis) is true. This includes the classification, testing, and rigorous examination of the hypothesis proposed in the third step. The fourth step uses induction and deduction to test the hypothesis. The fifth step uses those tests to evaluate sufficiency.

In practice, deduction, induction, and abduction are not proscribed to designated steps but rather recursively interplay to discover, describe, and test a hypothesis explaining a surprising fact. The transition between noticing a surprising fact and then producing a hunch is abductive, but this abduction is based upon existing knowledge structures produced by deduction and induction. The engendering of a hypothesis from this hunch requires repeated interactions of abduction with induction and deduction to adjust, test, and evaluate. Despite the natural amalgamation of the reasoning types within the retroductive process, the structure of these steps are useful to describe engineering design as a retroductive process.

Retroduction begins with the first step: the apprehension of a surprising fact, and design is no different. "The search for an explanation often begins when we notice that something is different, unusual, or wrong" (*Einhorn and Hogarth*, 1987). *Peirce* (1967) describes this as: "Every inquiry whatsoever takes its rise in the observation, in one or another of the three Universes, as some surprising phenomenon, some experience which either disappoints an expectation, or breaks in upon some habit of expectation."

The prompting of some phenomena differs according to discipline. A traditional example of this in scientific experimentation is realizing the orbit of a planet does not fit mathematical models and then hypothesizing that possibly the orbit is affected by another planetary body's gravitational field. The hunch stems from the scientist's knowledge of gravity and celestial bodies. The hypothesis is then tested and checked against other astronomical information.

In contrast to this, engineering design often requires the creation of new information rather than the exploration or application of existing information. The proclivity of designers to ask 'what-if' questions in early stage design to generate new analyses and information for potential designs is reflected in the prominence of tool features providing visualizations to support those inquiries. The Whole System Trades Analysis (WSTA) tool is one such tool and describes this interrogative process: "Because of WSTA's rich data base of subsystem choices and associated attributes, senior level stakeholders could ask 'what if' questions and explore the systems engineering trade space to obtain answers within hours and days rather than weeks and months." (*Edwards et al.*, 2015)

The second step of retroduction is the intuition of a hunch. A designer's hunch can manifest as a 'what-if' question. Tools are used to not just generate data but to assist in investigating these questions, often with visualization techniques. *Henry et al.* (2016) acknowledges this as "the [WSTA tool] results engine provides dozens of different filters and views with which to interrogate the resulting trade space." Interrogation of the design space is a hunch forcing the creation of new information in response to some prompting surprising fact or situation.

Hunches are often apprehensions of the designer's concern while searching for a surprising fact. Generally, formulation of a hunch is abductive reasoning as hunches surface due to intuition of the designer when confronted with a surprising fact or observation. These observations are neither good nor bad, just unexpected and can take the form of seeking out a novel design or as unanticipated phenomena. Experienced designers ask 'what-if' questions to intuit a hunch and construct a hypothesis in order to understand the surprising fact.

Step three of retroduction is the engendering of the hypothesis from a hunch. *Peirce* (1967) describes this as "by the aid of logical analysis . . . we convert 'B is heavy' into 'B has weight'." This stage is the formalization of the hunch into defined premises within a syllogistic form that can be examined with such abilities possessed by the reasoner. The process of formulating and defining premise are in themselves miniature reasoning instances informed by designer perspective and abilities. *Peirce* (1967) indicated this as "no doubt by introducing suitable definitions as premisses (sic) the same result can be reached syllogistically; but that is only because logical analysis has aided in the formation of those definitions." Once engendered, the hypothesis is tested as or step four of the retroductive process. From those tests, five begins evaluates the hypothesis for sufficiency.

There is significant interaction between the engendering, the testing, and the evaluation of the hypothesis. *Ammon* (2017) describes this interplay in design as "little by little, in hard-won steps and iterative loops, rightness is developed, options are checked countless times, revised, discarded or improved until, eventually, reliable knowledge emerges in a stabilized form which allows for the realization of the artefact." The cycle of retroduction requires an iterative refinement to continuously reshape the understanding of the surprising fact or inquiry that sparked the initial hunch and begins with that hunch being codified into a formal hypothesis.

Peirce (1967) notes that "retroduction does not afford security. The proposition must be tested." Testing of the hypothesis incorporates deduction and induction in support of sufficiency or the abductive generation of a hunch from a surprising fact. The hypothesis is defined in a context by the designer such that it can be tested. Testing often consists of examining data generated by a tool using the same tool's capabilities. This often proscribes the context of definitions. If the designer has generated data of resistance values from regression curve equations, the definition of 'good' in context must be predicated on resistance as that is what is available and accessible for testing purposes.

The testing of the hypothesis can consist of broad exploration of the solution space or a more targeted drill-down approach in an attempt to find an understanding of the answers to the "what if" questions asked. A more localised investigation is described in the context of the WSTA tool and shows how retroduction has an implicit reliance on expert designer opinion. *Henry et al.* (2016) describes a typical application of the WSTA tool for a new design program as one that requires "discussions with subject matter experts (SMEs)" within "an iterative refinement of the calculations used in the [functional objectives] based on further discussion with SMEs and data availability for the [technology options]." The iterative refinement mentioned is the generation and re-generation of data as the designer adjusts the objectives in pursuit of reaching some sufficient explanation to end this cycle of retroduction and advance to the next design activity.

Once tested, the hypothesis is evaluated for sufficiency. If testing does not prompt sufficiency the hypothesis is then revised. Revision of the hypothesis can consist of the inclusion of additional hunches or re-contextualizing the hypothesis for auxiliary testing. The new hypothesis is then interrogated and the cycle continues until a hypothesis is deemed sufficient to move on to the next design activity. At this point, the developed knowledge structures are codified as product structures for the next stage of design.

Peirce (1967) stated that "If Truth (sic) consists in satisfaction, it cannot be any actual satisfaction, but must be the satisfaction which would ultimately be found if the inquiry were pushed to its ultimate and indefeasible issue." Sufficiency is the truth in satisfaction of the hypothesis as the understanding behind a surprising fact that Peirce refers to here and is the eventual point at which the designer has collected enough evidence in support and can no longer improve her hypothesis. At that point, she has sufficient explanation and understanding behind the surprising fact that preempted the cycle of retroduction, and that individual cycle of retroduction can come to a close.

The evaluation of a hypothesis for sufficiency is consistent with designer's consideration of data generated by tool use in design activities. *Edwards et al.* (2015) emphasises the WSTA tool's role in ensuring that a designer "finally had access to sufficient data to make an informed decision."

The process of retroduction in logic theory is analogous to how designers use tools to draw conclusions for design decision-making. Therefore, this work uses logical syllogisms to articulate a novel framework for evaluating the suitable generation and realization of sufficient design data for decision-making. Details of this framework are presented in Chapter III.

CHAPTER III

Syllogistic Bayesian Framework for Evaluating Sufficiency

A retroductive framework for evaluating the suitable generation and realization of sufficient design data for decision-making is proposed in this chapter. The framework makes use of syllogisms (Section 3.1.1), Bayesian networks (Section 3.1.2), and mutual information (Section 3.1.3). Section 3.2 presents how the framework investigates decision sufficiency by evaluating decision statements as syllogisms by Bayesian networks. There are two possible network representations: a flat syllogistic instance network and a layered syllogism instance network. Associated metrics are probabilistic distributions and mutual information.

3.1 Background

Decisions are structured as logical syllogisms. These syllogisms are transformed into networks from which probability distributions and mutual information values are derived. The background components necessary to understanding the framework are contained in this section.

3.1.1 Logical Syllogisms

In order to examine the individual decisions that compose the retroductive design process, each reasoning instance of deduction, induction, and abduction are constructed as singular syllogisms. A syllogism is comprised of three statements: a major premise, a minor premise, and a conclusion statement. One syllogism has three total terms shared between statements. Each statement within the syllogism has two terms. The term shared between the major premise and the conclusion statement is the major term or the predicate term. The term shared between the minor premise and the conclusion is the minor term or the subject term. The term shared by the major and minor premise is the middle term.

Each statement is associated with a syllogistic mood which modify terms within the statement to be some strength of affirmative or negative expression. The modifier that does this is a result of the four syllogistic moods listed in Table 3.1. A traditional syllogism composed of three statements has three moods as each statement has a specific mood. U represents the first term in the statement and V represents the second term.

Mood	Statement
A	All U are V
E	No U are V
I	Some U are V
O	Some U are not V

Table 3.1: Syllogistic Moods (A,E,I,O)

In addition to a syllogism's three moods, each syllogism has an associated figure. Table 3.2 lists the four syllogistic figures as statement term pairs and denotes the predicate (major) term as 'P' , the subject (minor) term as 'S' , and the middle term as 'M'. Regardless of syllogism figure, the conclusion statement has the subject as the first term and the predicate as the second term.

A syllogism is described as 'XYZ-N' where 'XYZ' refers to the three moods se-

	Figure One	Figure Two	Figure Three	Figure Four
Major Premise	M-P	P-M	M-P	P-M
Minor Premise	S-M	S-M	M-S	M-S
Conclusion	All conclusion statements are S-P			

Table 3.2: Syllogistic Figures (1-4) by Statement

quentially: 'X' describes the mood of the first statement (the major premise), 'Y' describes the mood of the second statement (the minor premise), and 'Z' describes the mood of the third statement (the conclusion). 'N' describes the syllogism figure.

A example construction of an syllogism described by AAA1 is as follows. All three statements of AAA1 are in mood A: 'all U are V'. Terms from syllogistic figure one are substituted in for the terms U and V. Therefore, AAA1 is:

Major Premise: All M are P

Minor Premise: All S are M

Conclusion: All S are P

AAA1 is one of 256 argument forms, many of which are not useful for actual reasoning purposes. There are 24 syllogisms with logically valid forms, evenly split across the four syllogism figures. Of these, 15 are unconditionally valid (listed in Table 3.3) and 9 are conditionally valid (listed in Table 3.4). The definition of logical validity for syllogisms is a structure for which there is no premise configuration in which all are true such that the conclusion statement is False. Conditional validity requires the existence of some term for the form to be valid.

Figure 1	Figure 2	Figure 3	Figure 4
AAA	AEE	AII	AEE
EAE	EAE	IAI	IAI
AII	EIO	OAO	EIO
EIO	AOO	EIO	

Table 3.3: Unconditionally Valid Syllogisms

AAA1 is an unconditionally logically valid syllogism. However, that does not guarantee the truth of its conclusion. Should a logically valid syllogism be based upon

Required condition (must exist):		
Minor term	Middle term	Major term
AAI1	AAI3	AAI4
EAO1	EAO3	
AEO2	EAO4	
EAO2		
AEO4		

Table 3.4: Conditionally Valid Syllogisms

False or subjective premises, there is a risk the conclusion may also be False. The opposite of this also transpires. Syllogisms that are not logically valid can still result in True conclusions, even when based off of subjective premises. Erroneously confusing logical validity and actual truth can result in false confidence in this framework's value.

Each logical reasoning instance is associated with various syllogistic constructions. AAA1 is an example of an unconditionally logically valid syllogism construction of deduction. Forms with syllogistic figure one are generally deductive reasoning instances. Other syllogism figures are also associated with specific reasoning types. Forms with syllogistic figure three are generally inductive reasoning instances. AII3 is an example of an unconditionally logically valid syllogism construction of induction. Forms with syllogistic figure two or figure four are generally abductive reasoning instances. IAI4 is an example of an unconditionally logically valid syllogism construction of abduction.

Table 3.5 shows constructed syllogism statements for each logical reasoning instance.

	Deduction (AAA1)	Induction (AII3)	Abduction (IAI4)
Major Premise	all M are P	all M are P	some P are M
Minor Premise	all S are M	some M are S	all M are S
Conclusion	all S are P	some S are P	some S are P

Table 3.5: Example Syllogisms for Each Reasoning Type

Forms that are not logically valid are still useful to draw conclusions for decision-making. Many of these non-valid forms are those associated with abduction. The classical syllogism for abductive reasoning, AAA2, is not logically valid. (*Thompson*, 2016)

People tend to be comfortable drawing some conclusion regardless of logical validity. Unlike computers, designers can entertain contradictory or parallel ideas simultaneously. In part this can be traced to how the use of natural language modifiers such as 'some', 'none', and 'all' are open to semantic interpretations. What constitutes 'some' of a term may change depending on how a reasoner considers that idea in context of another. In order to quantify this, Bayesian networks are used to represent syllogisms probabilistically. Works that model probabilistic syllogisms treat moods A and I ('all' and 'some') and E and O ('not all' and 'some not') as structurally the same with differing term definitions affecting the probability distribution for that form. (*Hattori*, 2016; *Tessler and Goodman*, 2014)

3.1.2 Bayesian Networks

This section will highlight essential terminology and concepts of Bayesian networks for this work. For a comprehensive catalog of network theory, see *Newman* (2010).

Networks are representations of systems manifesting as a collection of nodes connected by edges. Generally, a node represents some determined unit and an edge represents a relationship between nodes. These edges can be directed or undirected depending on the desired network properties.

This work uses Bayesian networks to represent syllogisms because of the associated directional and probabilistic qualities. Each network is a directed acyclic graph in which a node corresponds to a variable with states and an edge corresponds to a directed, conditional dependency of nodal states. Simple examples of how Bayesian networks are constructed can be found in *Jensen and Nielsen* (2007).

Bayesian networks are probabilistic graphical representations that use Bayes' Theorem (Equation 3.1) for probability computations. Equation 3.1 defines the probability of event X occurring given knowledge of event Y, to be equal to the ratio of the probability of both events X and Y occurring to the probability of event Y.

$$P(X|Y) = \frac{P(X \cap Y)}{P(Y)} = \frac{P(Y|X) * P(X)}{P(Y)} \tag{3.1}$$

Equation 3.1 is also defined in terms of conditional probability. The notation $P(X|Y)$ describes the set of conditional probability distributions, each associated with nodes in a Bayesian network. These distributions can be presented as conditional probability tables (CPTs) which contain the statistical distribution of nodal states within the network. Examining these distributions assists in development of insights into what and how information can affect decision-making. Note that the conditional probability is not defined when $P(Y) = 0$, resulting in an undefined value when evidence does not exist.

Syllogisms are transformed into Bayesian networks so as to examine sufficiency. Bayesian Networks provide a structured way to look at the probabilistic causal relationships inherent in and between syllogism statements (*Pearl*, 1988). Each node of the Bayesian network represents a part or parts of a syllogism. Edges represent the ordering of decision statements. This work uses two representations of a syllogism as a Bayesian Network: a flat and a layered network. Defining nodal states is associated with statement component values and is detailed fully later in this section.

The utilization of semantics requires a discussion on the difference between logical term values and nodal state values. Two semantically different statements are: 'I'm not *not* hungry' and 'I'm hungry.' These statements are equivalent when represented in propositional logic as $\neg(\neg H) = H$, where H stands for 'hungry'. However, a reasoner knows that the emphasis on the second 'not' gives these two statements different interpretations.

Semantic and propositional logic values also differ on the equivalence of $A+B$ and $B + A$. Propositional logic considers addition of terms or premises as a symmetric process, However, knowledge structures are path dependent which gives either A or B emphasis (*Page*, 2006). This arises as a concern for differentiations between the minor and major premises. Syllogisms have an inherent premise and term prioritization which is not reflected in propositional logic. Ordering matters for premise contribution to a conclusion. Often a design is disproportionately influenced by early knowledge incorporation.

Situations similar to this require an adjusted mathematical interpretation during the transformation between the logical term value and the nodal value so as to not lose their semantic implications.

Syllogism logic traditionally uses Boolean functions as logical values. This can divorce statements from significant semantic implications. A designer can classify information beyond just True or False.

Several novel methods were developed in this work to account for the transformation of semantic language to the logical value of terms within a syllogism and then to a probability distribution for a Bayesian network. First, the syllogism term component values as relating to nodal values include a state indicating non-relevance in addition to True and False. Secondly, a method for a layered network was developed and is presented to account for premise ordering implications.

From the flat and layered networks, probabilities are calculated and presented as CPTs. Mutual information for both types of networks is also calculated.

Flat Bayesian network

A singular syllogism can be represented as a simple three-node Bayesian network. Each node represents one statement of the syllogisms. The edges connect the statements following the logical flow of the syllogism's structure: major premise to minor premise, major premise to conclusion, and minor premise to conclusion (seen in Figure 3.1). This network representation is called flat due to the premises represented as separate nodes and on the same plane of the network.

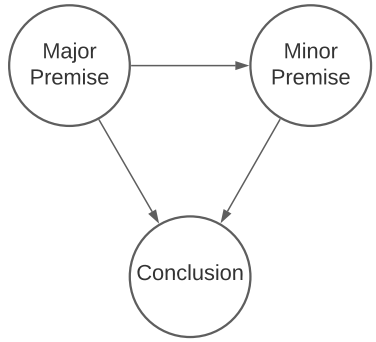

Figure 3.1: Bayesian Network for Flat Syllogism Representation

Nodal state values can be True, False, or not applicable (N/A). The value of including both a False and N/A state is to account for the preeminence of the first term within a statement. Semantically, 'if A, then B' can have three possible outcomes for evidence. If A provides a context in which a designer is interested in, then the statement is True or False depending on if B is True or False. But if A does not do so, that does not mean the statement is False.

Lack of qualifying evidence (N/A) is not inherently contradictory to a statement the same way a False state is. By including an N/A nodal state, this work seeks to incorporate some small amount of uncertainty in nodal value statements appropriate to their original semantic form, approaching the way a pseudo-fuzzy logic approach might.

Since each node represents a statement, the process for determining the state for a node is tied back to the component value of the two syllogism statement parts. A statement can be constructed into two parts (X_0-X_1) where X_0 is the first term and X_1 is the second, each consisting of the modifier from the syllogism mood and term.

This can be seen using the example statement "if on Main Street, eat at a fun restaurant". The first part of the statement (X_0) would be 'Main Street'. The second part of the statement (X_1) would be 'fun restaurant'. The modifier words are both positive. Modifier words in partial statement component values are included in case of term negation. Nodal state values are assigned based off of component values of the partial statements X_0 and X_1 as seen in Table 3.6. component values for partial statements come from term definitions set by the reasoner.

X_0 Value	X_1 Value	Nodal Value
T	T	T
T	F	F
F	T	N/A
F	F	N/A

Table 3.6: Partial Statement Values Relating to Nodal State Values for Flat Network

Nodal values of N/A occur when a component value of False for the first part of a statement indicates that the information contained within the associated piece of evidence is not relevant. Say a reasoner is considering the statement used above: "if on Main Street, eat at a fun restaurant". The example evidence statement, 'On Ashley Street at Fleetwood Diner', has a first part (X_0) component value of False; Ashley Street is not on Main Street. This piece of evidence is thus non-relevant information and the nodal value is labelled N/A.

An example of a piece of relevant evidence is 'On Main Street at the Chop House'. The primary term is Main Street (X_0 value True) so it is considered relevant. The nodal state is then tied to the component value of the secondary part of the statement while the first part determines the relevance.

What determines the component value of partial statements is context dependent. To determine the component value of the secondary statement requires consideration of what is a 'fun' place and how can it be quantified. Obviously, 'fun' is context dependent. Perhaps the reasoner defines 'fun' as restaurants that serve alcohol. Since the Chop House serves alcohol, it has a True component value. Thus, the nodal state is likewise True.

Context can define 'fun' differently. Someone who does not care for drinking may consider 'fun' to be an extensive mocktail list instead.

Any quantification of semantics requires a reliance on a definition. 'Fun' could also refer to places that have happy hours or outdoor seating or anything the reasoner deems fitting and matches their perspective of 'fun'. However, term definition is also tied to availability of information. For example, if the reasoner doesn't know what restaurants have happy hours, then that cannot be used to define 'fun.'

Layered Bayesian network

Flat networks and their associated probability distribution and mutual information values allow for evaluation of a syllogism with a clear prioritization of the major premise over the minor premise. However, flat networks lack understanding of intra-premise relationships. A layered approach was developed in order to examine context to context relationships.

This approach removes the subjectivity of the designer determining what is the major and minor premise. Within logic theory, there are no mechanisms to determine what is what within the syllogism structure. Often, the determination of what is the major premise and what is the minor is speculation or conjecture. If the reasoner were to select inappropriately, it can result in confusion. The designer can really can only determine what is of interest, which is represented in the selection of the terms within the syllogism.

The layered network utilizes a premise node that is comprised of the major and minor premise instances. This singular node's value is based on the mechanisms on which syllogisms work: the terms that make up the syllogism statements. In this approach, the syllogism terms make up a singular node that represents the premises within a syllogism.

Edges between nodes show the flow of information from the combined premise statements to the conclusion statement. Multiple grouped premise statements can contribute to a single conclusion. These grouped premises have a subhierarchical order. For convenience, statements included in this node are described as a major premise and a minor premise. Figure 3.2 shows the layered network.

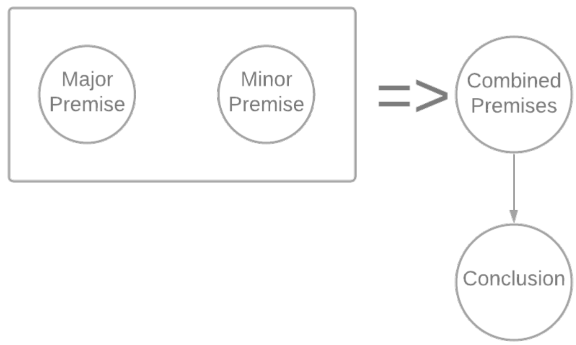

Figure 3.2: Bayesian Network for Layered Syllogism Representation

Similar to the flat Bayesian network, nodal states of the layered network have possible values of True, False, or N/A. Relevancy for the flat network is determined by the first term of the statement since the nodes represent singular statements. For the layered network, relevancy is determined by the middle term. The middle term connects the major and minor premises. Within the combined premises node it sets the context of the relationship necessary for decision-making. This context must exist; that is, the middle term has a value of True. If the middle term has a value of False, that indicates that the lack of connection between the two premises prevents the evidence's use in decision-making. The connection between premises is determined by a True value that indicates the reasoner's belief in the term and thus association of the premises. Premises with a True nodal states are not indicative of actual correctness but rather an affirmation of the designer's construction of those premises definitions.

Unlike the flat network, the layered network represents reasoning instances associated with syllogism figures with distinct translations of statement term values to nodal states. Table 3.7 shows a archetypal syllogism defining each term and position within the three syllogism statements.

P_i refers to the two terms of the major premise: the predicate term and the middle term. The subscript refers to the position as either the first term (P_0) or the second (P_1). S_i refers to the two terms of the minor premise: the subject term and the middle term. Which terms are the middle term depends on the syllogism figure.

Major Premise:	P_0 - P_1
Minor Premise:	S_0 - S_1
Conclusion:	S - P

Table 3.7: Archetypal Syllogism Term Definition

Table 3.8 defines terms for each syllogism figure. Deduction, induction, and abduction can all be represented within this format by different definitions of $P_{i=0,1}$, $S_{i=0,1}$, and M as the predicate, subject, and middle terms respectively. Recall that syllogism figure one is deduction, syllogism figure three is induction, and syllogism figures two and four are abduction.

Term	Figure One	Figure Two	Figure Three	Figure Four
Predicate	P_1	P_0	P_1	P_0
Subject	S_0	S_0	S_1	S_1
Middle	P_0, S_1	P_1, S_1	P_0, S_0	P_1, S_0

Table 3.8: Syllogistic Figures (1-4) by Term

Nodal state value tables for syllogism figures one, two, and three are seen in Table 3.9, Table 3.10, and Table 3.11 (deduction, induction, and abduction respectively). Bolded nodal values vary between reasoning types depending on ordering of the predicate and subject within the major and minor premises for reasoning types. Term component values are True or False and nodal states for the layered network can be True, False, or not relevant (N/A), same as the flat network nodal values.

The assignment of nodal values for deduction, induction, and abduction are identical except for the bolded situations. As mentioned previously, when the middle term is False, the premise connection is not relevant and thus the nodal status is N/A. When the middle term is True, the combined premises are relevant. In this circumstance, if the predicate and subject terms are both the same component value (True

M ($P_0 = S_1$) Value	P_1 Value	S_0 Value	Nodal Value
T	T	T	T
T	T	F	**F**
T	F	T	**T**
T	F	F	F
F	T	T	N/A
F	T	F	N/A
F	F	T	N/A
F	F	F	N/A

Table 3.9: Syllogism Figure One (Deduction) Values Relating to Nodal State Values for Layered Network

M ($P_0 = S_0$) Value	P_1 Value	S_1 Value	Nodal Value
T	T	T	T
T	T	F	**T**
T	F	T	**F**
T	F	F	F
F	T	T	N/A
F	T	F	N/A
F	F	T	N/A
F	F	F	N/A

Table 3.10: Syllogism Figure Three (Induction) Values Relating to Nodal State Values for Layered Network

or False), the nodal value corresponds to the same value as the predicate and subject. Reasoning type affects the determination of the nodal value when the predicate and subject disagree.

The major premise consists of the predicate term and the middle term. The first term of a statement is the context driving term. When the first term of the major premise is the predicate (P_0) with a False component value, the nodal state is likewise False. This holds for abductive syllogisms where the predicate is the first term in the major premise. Context is especially important in abduction which attempts to provide insight from circumstances into statements. Falseness here means a lack of context and thus flawed decision-making.

Deduction and induction have the predicate term as the second part of the major premise (P_1). When the predicate has a False component value and is the second

M ($P_1 = S_1$) Value	P_0 Value	S_0 Value	Nodal Value
T	T	T	T
T	T	F	**F**
T	F	T	**F**
T	F	F	F
F	T	T	N/A
F	T	F	N/A
F	F	T	N/A
F	F	F	N/A

Table 3.11: Syllogism Figure Two (Abduction) Values Relating to Nodal State Values for Layered Network

term in the major premise, the nodal state is tied to the subject component value and position in the minor premise. When the subject term is also False, the nodal state is False. When the subject term's component value is True, the nodal state depends on the location of the subject term in the minor premise. When the subject is the first term (S_0) as in deductive reasoning, the nodal state is True. The subject as the first term of the minor premise (S_0) drives context more than the predicate as the second term of the major premise (P_1), and thus the nodal state is True to match the subject. In inductive reasoning, the subject of the minor premise is the second term (S_1) so the nodal state is False to follow the component value of the predicate. This is to express that the major premise is assumed to be more of a context driver than the subject in the minor premise.

Regardless of the position of the predicate term in the major premise, when the predicate has a True component value, the nodal state depends on the subject term's component value and position. If the subject is also True, then the nodal state is True. If the subject is False, then the nodal state depends on the location of the subject term in the minor premise.

In deductive and abductive reasoning, the subject is the first term of the minor premise (S_0). When the subject is False, the nodal state is False. The first term of the minor premise drives context more than the predicate as the second term (P_1), and thus the nodal state is False to match the subject.

In inductive reasoning, both the predicate and subject are second terms in the major and minor premises (P_1, S_1). When the subject is False and the predicate is True, the nodal state is True, following the component value of the predicate.

As mentioned previously, syllogisms structured as figure four are abductive reasoning instances. These figures are not included in this framework due to their cyclic nature and the difficulty of representing that as a directed network. The syllogism structure of figure four is such that the term determination is dependent entirely on syllogism statement ordering. Table 3.12 shows three examples of figure four syllogisms with identical statements but different predicate, subject, and middle terms.

A-B	B-C	C-A
B-C	C-A	A-B
C-A	A-B	B-C

Table 3.12: Syllogism Figure Four Examples

Presumption of premise ordering would determine nodal values without context due to the several possible configurations. A, B, or C could all be the middle term and determine relevancy.

In order to eventually adjust this framework to include figure four as a syllogistic representation of abduction, nodal states would need to be expanded to adjust to the loose context and conditionality inherent in figure four's syllogism structures.

3.1.3 Mutual Information

Decision-making requires an assumption of connection between premises and conclusions. Often there are hidden interdependencies or other biasing within these statements. In order to quantify and evaluate these relationships, mutual information is derived.

Mutual information measures the information that events can share. It is a dimensionless quantity with defined units of nats (the natural unit of information) and can be thought of as the reduction in uncertainty about one event given knowledge of another. A designer seeks to mitigate risk by reducing uncertainty, and so this metric allows for a quantifiable reduction in risk to be compared depending on construction of a decision.

Mutual information is intimately tied to entropy, the disorder of a system. This work uses mutual information instead of entropy as mutual information more easily is calculated from generated probability distributions. Different methods to represent syllogisms may prefer different metrics.

Mutual information between two jointly discrete random variables, X and Y, is defined by Equation 3.2, where $p(x, y)$ is the joint probability mass function of X and Y and p_X and p_Y are the marginal probability mass functions of X and Y respectively.

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \ln \frac{p(x, y)}{p_X(x) p_Y(y)} \tag{3.2}$$

Mutual information, $I(X;Y)$, is a measure of the inherent dependence expressed in the joint distribution of X and Y relative to the marginal distribution of X and Y under the assumption of independence. Mathematically, mutual information is a measure of how the joint distribution of two random variables deviates from the case where they are independent of each other. Mutual information therefore measures dependence in the following sense: $I(X;Y) = 0$ if and only if X and Y are independent. For any amount of dependence, $I(X;Y)$ is non-zero. $I(X;Y)$ cannot be negative.

Mutual information among three variables is called the interaction information and can be calculated by Equation 3.3. (*Tsujishita*, 1995)

$$I(X;Y;Z) = \sum_{z \in Z} \sum_{y \in Y} \sum_{x \in X} p(x, y, z) \ln \frac{p(x, y) p(x, z) p(y, z)}{p(x, y, z) p(x) p(y) p(z)} \tag{3.3}$$

Equation 3.4 shows interaction information, $I(X;Y;Z)$, presented in the form of mutual information, $I(X;Y)$, and conditional mutual information, $I(X;Y|Z)$. Conditional mutual information is the expected value of the mutual information of two random variables (X and Y) given a third (Z).

$$I(X;Y;Z) = I(X;Y) - I(X;Y|Z) \tag{3.4}$$

$I(X;Y;Z)$ measures the influence of a variable Z on the amount of information shared between X and Y. When the interaction information is negative, it indicates that $I(X;Y|Z)$ is larger than $I(X;Y)$. This will happen when X and Y are independent but not conditionally independent given Z. $I(X;Y;Z) > 0$ indicates that Z inhibits (i.e., accounts for or explains some of) the correlation between X and Y, whereas $I(X;Y;Z) < 0$ indicates that variable Z facilitates or enhances the correlation. Negative interaction information is called synergy, indicating that variables taken together are more informative than when they are taken separately (*Schneidman et al.*, 2003).

Mutual information is symmetric; that is $I(X;Y) = I(Y;X)$. This is also true for interaction information: $I(X;Y;Z) = I(Y;Z;X) = I(Z;X;Y)$, etc. This means that the mutual information value is identical no matter which variable it is conditioned on.

The symmetry of mutual information leads to a desire for another way to understand the combined affect of major and minor premises on the conclusion nodal status.

There is not a goal or a threshold mutual information value to indicate sufficiency. Like the way designers make decisions, cases are compared and evaluated depending on specific circumstances of the knowledge development or queries articulated and quantified as these values in comparison to other syllogisms examined. Interpretation of mutual information in the context of the case is presented in Chapter IV.

Flat Network Mutual Information

The flat network structure includes separate nodes for the three syllogism statements: the major premise, the minor premise, and the conclusion. Interaction information, as seen in Equation 3.3, is used to calculate the flat mutual information. Interpretation of $I(X;Y;Z)$ is below:

$I(X;Y;Z) = 0$: All statements are independent

$I(X;Y;Z) > 0$: One of the statements accounts for some of the relationship between the other two.

$I(X;Y;Z) < 0$: One of the statements magnifies the relationship between the other two.

Layered Network Mutual Information

The layered network structures a combined premise node out of the major and minor premises. Equation 3.2 is used to calculate the mutual information, called layered mutual information: $I(XY;Z)$.

Interpretation of $I(XY;Z)$ is below:

$I(XY;Z) = 0$: The combined premises and the conclusion are independent.

$I(XY;Z) > 0$: Premises and conclusions are correlated and thus potentially better for decision-making.

$I(XY;Z) < 0$: Premises and conclusions have large amounts of non-relevant evidence. Connection between premises and conclusions is unclear and cannot be determined for decision-making.

3.2 Evaluating Decision Sufficiency

Retroduction in engineering design can be structured as seen in Figure 3.3. This five-step process provides a convenient way to describe the framework.

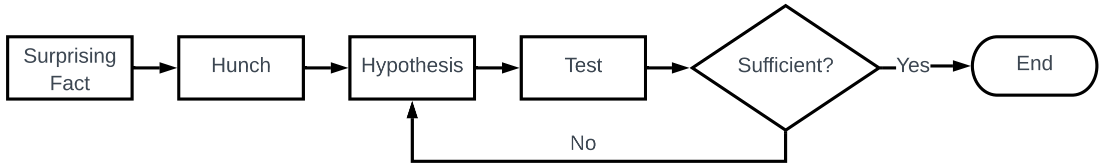

Figure 3.3: The Retroductive Process as a Series of Steps

In order to evaluate the decision sufficiency, first a tool must be used to generate data for knowledge development. The Sen bulk carrier code (Appendix A) is one such tool. Tools such as this one are often considered suitable for a wide range of use without critical examination of the generated data for decision-making. This framework allows for the evaluation of design inferences by utilizing logical syllogism constructions of design space exploration data as the means for the determination of sufficiency of generated data used for decision-making.

Step 1: Apprehension of a surprising fact

Experienced designers pose 'what-if' questions to seek out surprising facts in anticipation of discovering some novel aspect. These 'what-if' questions interrogate the data sets generated by the tool to develop the initial knowledge structures. The apprehension of a surprising fact can arise from any notification or notice of something that stands out in relationship to a design activity. For this framework, that requires the creation of evidence as a codification of the knowledge structure.

Table 3.13 is an example of a selection of evidence from a modified Genetic Algorithm (GA) developed based on the Sen bulk carrier code. Each row is a run for a specific tournament size, with associated numbers of solutions. 'PF ever' and 'PF final' refer to the number of solutions generated during a run of the GA ever on the Pareto front and the number of solutions on the Pareto front at the final epoch for a run of the GA.

Generation and initial perusal of this evidence prompts the designer to begin the

tournament sizes	total solns	feasible solns	PF ever solns	PF final solns
2	815	742	35	7
3	830	721	33	9
5	862	609	62	15

Table 3.13: Example Evidence from Sen Bulk Carrier Code

formation of certain understandings that may eventually solidify into premises for decision-making. This begins step 2, the intuition of a hunch.

Step 2: Intuition of a hunch

From the initial examination of tool-generated data sets, an intuition of a hunch arises. Abductive connections are made into the 'why' of the surprising observations in the form of answers to a 'what-if' question. A designer may ask several 'what-if' questions in pursuit of answers as to the 'why' behind the surprising fact.

The conceptualization of a hunch requires the selection of concepts or areas that the designer can form terms to construct syllogisms from. A designer using the Sen bulk carrier tool may begin thinking about quality of solutions generated by the optimizer.

Step 3: Engendering of the hypothesis

Once a hunch arises, it must be constructed in a testable way. This requires a formalization of the hunch into a hypothesis. Concepts of interest that were associated with the hunch in step 2 are rigorously defined as terms such that they are testable.

For example, a designer whose hunch is associated with the quality of the Sen bulk carrier optimizer may add nuance to the concept of quality. How can quality be defined? She may consider quality to be the number of solutions who at any time were located on the Pareto front. She could also define it in other ways such as in terms of feasible solutions, Pareto front final solutions, or total solutions. All of these potential definitions are dependent on what 'what-if' question and hunch the designer is considering.

Data must be organized or quantified the same way as the syllogism statements are. In design, this happens organically as a designer matches her motivations and perspective of ambiguous statements such as ‘enough’ or ‘good’ solutions to arbitrary quantitative delineations for statement definitions. Definitions depend on designer’s semantic understanding of the language used within the syllogism statements to translate qualitative into quantitative evidence.

Defined terms are arranged into a hypothesis that is constructed as a syllogism. The syllogism can be constructed several ways depending on the designer’s hypothesis and perspective. Details on how syllogisms are constructed are in Section 3.1.1.

Step 4: Testing of the hypothesis

To gain insight into the sufficiency of the generated data sets for decision-making, the relationships between the syllogism statements must be examined. From the syllogism or syllogisms formed in step three, a flat and a layered network are constructed. Networks provide a structure that can determine dependence among nodal representations. A network is case-dependent and can represent a singular syllogism engendered from a hypothesis or multiple syllogisms from the multiple hypotheses examined during the retroductive cycle. Section 3.1.2 details how the Bayesian networks are generated.

Probabilities and mutual information values are calculated from both flat and layered networks. To produce the probability distributions, each syllogism statement has already been defined in context of the evidence that will be used to populate the networks. These calculated probabilities will be used to calculate the mutual information values as seen in Section 3.1.3.

Step 5: Evaluation of sufficiency

Relationships between syllogism statements contributing to a conclusion for decision-making are examined for insights into sufficiency of generated data.

Sufficiency of generated data used for decision-making is evaluated by calculating and comparing the quantified values of probabilities and mutual information for flat and layered network syllogism representations.

Probability distributions allow for an examination of the strength of statements relationships. Drawing a conclusion requires premises to impact one another. These distributions show overlap between statements.

Mutual information measures how much a variable can inform about another variable. Knowing the reduction in uncertainty a statement can bring can mitigate risk in decision-making.

Certain syllogism statements proposed in the hypothesis are tested in a comparative manner to quantify their conditional dependencies and reduction in uncertainty on each other.

Sufficiency of generated data for decision-making depends on the strength and type of syllogism statement relationships. Should the evaluated metrics not indicate a suitable hypothesis, the third step is returned to and the hypothesis is re-engendered. Statement and term definitions may also be revised to take into account new knowledge developed during the retroductive process. Then the fourth and fifth steps are repeated, with the new hypothesis tested and then evaluated for sufficiency. Hypothesis formulation, testing, and evaluation can be repeated multiple times as needed until the designer is satisfied and confident in her sufficiency judgement.

CHAPTER IV

Decision Sufficiency Case Study

A representative case study was developed to demonstrate the application of the retroductive decision-making framework. The case exemplifies the retroductive process initiated by a designer asking 'what-if' questions during an early stage design activity.

To show the application of the framework described in Chapter III, this chapter will examine several subjects that affect the suitability of tool-generated data for decision sufficiency:

1. Comparison of flat and layered constructions.

2. Implications of the use of deductive, inductive, and abductive syllogisms.

3. Assignment of terms for hypotheses.

An overview of the case's retroductive process is described in Section 4.1. Verification and validation of the tool-generated datasets used for reasoning is delivered in Section 4.2. The syllogisms and associated terms constructed and defined are in Section 4.3. The results of applying the syllogistic and network methodology to the case are presented in Section 4.4. Sufficiency is discussed in context of the case in Section 4.5. Case conclusions are presented in Section 4.6.

4.1 Story

A designer has been tasked with determining preliminary dimensions for a bulk carrier in early stage design. The tool available for this task is an optimizer based on the Sen bulk carrier code. She knows that the optimizer is based off of poorly-documented code historically used to generate bulk carrier designs. She has concerns that she may use the tool in such a manner as to generate data unsuitable for drawing inferences for knowledge generation and decision-making.

Her concerns stem from knowing that the original code has certain component values based on regression equations. The optimizer she is using produces dimension values to the millionth decimal place. She thinks that because of this over-precision, she may be able to truncate certain aspects of the code to speed up run time, without compromising the quality of the generated solution data or knowledge generated from use of this tool for decision-making.

Is the data generated by her tool use suitable for decision-making? Will it allow for drawing sufficient conclusions? The framework detailed in Chapter III allows the designer to evaluate suitability and sufficiency. This section will detail her methods and motivations in the context of the retroductive decision-making process to show how she determines this. Specifics as to the use of the tool are detailed in Section 4.2.

Step 1: Apprehension of a surprising fact

The designer wants to know if generated solution data produced by the optimizer is useful should she truncate the code for the optimizer. In order to develop knowledge structures for future decision-making, she needs to understand the relationship between truncation and quality of solutions. The designer does not want to draw conclusions for decision-making off of non-suitable design data.

Her question of 'what-if' she truncates the code is prompted from the surprising fact of the overly precise dimensions of the unmodified code. The observation that the dimension values are precise to the millionth decimal place is surprising to her as she believes that the shipyards that will eventually build the design she is developing will not be working to a length dimension that specific. She also suspects that the regression equation the code was based off of did not have the same degree of precision as the new optimizer.

These observations of the unnecessarily precise dimensions in congruence with the designer's desire to truncate code aspects compels designer consideration. She wishes to understand if solution data generated by the tool when truncated would allow her to form conclusions for decision-making.

Step 2: Intuition of a hunch

Despite the designer's concerns, she hopes that truncating certain optimizer aspects will still generate solution data useful for decision-making. Her hunch is that the type of truncation will affect the quality of the solution data in regards to a reduction of diversity or the quantity of unique solutions. Thus, she wants to examine different types of truncation in order to make an informed choice for what generated data is suitable to use.

The designer is concerned that there may be unobserved dependencies or holdovers from older versions of this optimizer code that will affect generated solution data. She specifically wants to make sure that a truncation code can produce enough solutions of a certain quality for her to acquire representative knowledge.

Step 3: Engendering of the hypothesis

In order to test and evaluate the designer's hunch that different types of truncation may change the relative quality and quantity of solutions from the optimizer, terms must be defined and formalized as syllogisms. The selection of these terms arises from the designer's thought process described in steps one and two. In summary, the designer wants to truncate values within the optimizer. Her concerns about quality and quantity of solutions suggests the following terms to be defined and examined: 'more truncation', 'useful optimizer', tied to her perception of quality, and 'enough solutions', tied to her perception of quantity.

She is primarily concerned about the production of enough solutions.

How the designer defines 'useful optimizer', 'more truncation', or 'enough solutions' impacts the evaluation of tool appropriateness and conclusion sufficiency. Each term is quantitatively defined in the context of the generated data that is used for testing and analysis.

There are 18 possible syllogisms that can be engendered from these three terms. Section 4.3 describes the syllogism constructions relevant to the designer's purpose and how their definitions were developed for this case.

Step 4: Testing of the hypothesis

From the syllogisms and associated term definitions developed in step three, networks are created and populated by evidence to produce probability distributions and mutual information values. Both flat and layered networks and their associated metrics are created for each syllogism. For this case study, the designer runs the optimizer to generate a large solution space populated with potential solutions created with varying types of truncation.

Evidence to produce the probability distribution and mutual information metric values for nodes in the Bayesian network are the runs described within the datasets in Section 4.2.

Step 5: Evaluation of sufficiency

Once the syllogism networks have evaluated the probabilities and mutual information values, the designer must evaluate the hypothesis for sufficiency of generated data for decision-making. The tested syllogism metrics allow for a quantification of designer subjective perspective.

Depending on the evaluation, the conclusion may be sufficient for decision-making and understanding of design tool-generated data suitability. Or, stage three is returned to for a revised hypothesis and repeated testing for additional evaluation.

Sufficiency is context dependent. This framework enables the evaluation of design inferences utilizing logical syllogism constructions based off of designer perspective of design space exploration data. The designer can determine sufficiency of the generated data used for knowledge queries.

4.2 Dataset Generation and Validation

This section details certain assumptions required to create and test the network syllogism representations. The optimization code, data generation, and validation are discussed to provide objective comparisons with case results.

Datasets used for the networks were generated from the modified Genetic Algorithm (GA) that *Sypniewski* (2019) created of the bulk carrier synthesis model from *Sen and Yang* (1998) (detailed in Appendix A).

To create each dataset, the modified GA was run with default parameters as detailed in *Sypniewski* (2019) for the baseline objective function and tournament size of 2. The model was run without truncation and with seven combinations of truncation categories (Table 4.1), creating eight datasets. Each dataset consists of many runs with differing decimal truncations rounding the designated component values to 1, 2, 3 and 4 decimal digits.

For labelling purposes, ‘Input’ truncation refers to truncating ship dimension values when calculated within the GA. ‘Penalty’ refers to truncating the penalty function when calculated. ‘Constraint’ refers to truncating the bulk carrier constraint equations. For notation, if the truncation category is labelled ‘Input’, it should be assumed that ‘Constraint’ and ‘Penalty’ are not truncated.

Truncation categories
Input
Penalty
Constraint
Input and Penalty
Input and Constraint
Penalty and Constraint
Input, Penalty, and Constraint

Table 4.1: Truncation Categories for Evidence Generation

All data and meta data from individual runs were saved to files. Table 4.2 lists saved variables and associated descriptions. A run consists of many solutions. Types of solution variables within a run are as follows: solution ID, penalty, feasible status, epoch start, epoch end, Pareto front epoch start, Pareto front epoch end, Pareto front ever, Pareto front end, length, beam, draft, depth, speed, block coefficient, transportation cost, lightship mass, annual cargo, generation total runs, generation tournament size, generation population size, generation stopping epoch, generation identification, crossover type, crossover probability parameter, crossover alpha parameter, run number, parent 1 ID, and parent 2 ID. For in depth descriptions of how these variables are calculated, refer to *Sypniewski* (2019).

Any data or information from the optimizer can be used to quantitatively define syllogism terms. Evidence of interest for this case is listed in Table 4.3.

‘Run ID’ is the overall identification number or runs within truncation category datasets. ‘Seed’ is the number used to initiate the random number generator so runs can be compared for the same initializations. Truncation category refers to

Data Category	Additional Descriptor
Dataset	n/a
Seed Number	n/a
Run ID	n/a
Tournament Size	n/a
Truncation	Decimal, Category
Number of ... Solutions	Total, Feasible, Pareto-ever, Pareto-final
Cardinality	Feasible Cumulative, Feasible Not Cumulative, Pareto Cumulative, Pareto Not Cumulative
Generational Distance	Superfront Value, Convergence
Distance from Superfront	Feasible, Pareto
Distance to Superfront	Feasible, Pareto

Table 4.2: List of Data and Metadata Recorded for Each Run of the Model

Run ID	Seed	Truncation Category	Decimal Truncation	Total Soln	Feasible Soln	PF Final Soln
27	3	all3_truncated	1	826	696	11
10305	25	penalty_truncated	4	813	556	95

Table 4.3: Example Values for Evidence

those listed in Table 4.1. Decimal truncation is the number of decimal places the truncation categories are truncated to. 'Total Soln', 'Feasible Soln', and 'PF Final Soln' each respectively refer to the number of total number of solutions generated during the run, number of feasible solutions within a run, and number of solutions on the final Pareto front at the end of the run.

To validate truncated datsets with each other and with unaltered datasets, the average distances to and from the superfront of the Pareto set are calculated and compared. The unaltered dataset cannot form a syllogism and be evaluated with other case metrics. Therefore in order to have a measure of comparison with an unaltered dataset, alternative methods must be used.

A superfront is the Pareto set of Pareto sets. It contains the overall dominate solutions from all runs generated for all datasets, truncated or not. Equations 4.1 and 4.2 state the formulas for distance to the superfront, $D_T(A)$, and from the superfront, $D_F(A)$, respectively. A is the solution set and SF is the superfront set. *Sypniewski* (2019) provides additional detail of the superfront and calculations.

$$D_T(A) = \frac{1}{|A|} \sum_{i \in 1}^{n} \min_{j \in SF} d_{ij} \tag{4.1}$$

$$D_F(A) = \frac{1}{|SF|} \sum_{i \in 1}^{n} \min_{j \in A} d_{ij} \tag{4.2}$$

Table 4.4 lists the calculated average distance to the superfront for each suite of runs generated for a decimal truncation and category. Large average distances of datasets to the superfront indicate optimal solutions of that dataset are worse than those with smaller distances.

Truncation Categories	Digits Truncated	Distance To Superfront (Pareto)	Scaled (%)
No Truncation	**X**	**0.02412**	**0**
Penalty, Constraint	3	0.02467	0.55
Constraint	4	0.02474	0.61
Input, Constraint	4	0.02525	1.11
Constraint	2	0.02552	1.38
Penalty, Constraint	4	0.02568	1.53
Input, Penalty	4	0.02569	1.54
Input	4	0.02595	1.8
Penalty	4	0.02624	2.08
Constraint	3	0.02649	2.32
Input, Penalty, Constraint	3	0.02662	2.45
Input, Penalty, Constraint	4	0.02676	2.59
Input, Constraint	3	0.02693	2.75
Penalty	2	0.02705	2.87
Penalty, Constraint	2	0.02706	2.88
Penalty	3	0.0276	3.42
Input, Penalty	3	0.02771	3.52
Input	3	0.02975	5.52
Penalty	1	0.0314	7.13
Constraint	1	0.0319	7.62
Penalty, Constraint	1	0.03342	9.11
Input	2	0.03641	12.03
Input, Penalty, Constraint	2	0.03976	15.32
Input, Constraint	2	0.04012	15.67
Input, Penalty	2	0.04037	15.91
Input, Penalty, Constraint	1	0.1166	90.55
Input	1	0.12143	95.27
Input, Constraint	1	0.12616	99.9
Input, Penalty	1	0.12626	100

Table 4.4: Average Distance To Superfront (Pareto) for All Datasets

Table 4.5 lists the calculated average distance from the superfront for each suite of runs generated for a decimal truncation and category. Larger average distances of datasets from the superfront indicates a worse optimality of the solutions of that dataset as well as less solution diversity. The smaller the distance, the better a dataset is with less homogeneous solutions.

The average distances to and from the superfront allow for a comparative examination of the quality of optimizer solutions as defined in relation to Pareto-set solutions when truncated as such. It does not allow for determination of sufficiency or incorporation of quantity of solution concerns without much extrapolation on behalf of the designer. It is intuitive that categories truncated to fewer digits (1,2) should be further to and from the superfront (have worse optimal solution sets) than those truncated to more digits (3,4, and X, which indicates no truncation), all other things assumed equal.

This is seen in Table 4.4, which is organized in ascending order. The scaled column specifies the percent difference between the corresponding distance and the smallest value scale over the range of distance values. Unsurprisingly,the solution set generated without truncation is located with the smallest values at the top (bolded). As expected, several 1 and 2-digit truncations are located in the bottom half, having large average distances to the superfront and indicating that more truncated solution sets have less optimal solutions than less truncated solution sets.

Should all truncation categories be assumed equal, the expectation would be that the table should have an evenly distributed array of truncation categories throughout, differentiated only by digits truncated. However, those truncation categories that include 'Input' cluster at the bottom of the table, with large average distance to the superfront indicating their comparative lack of Pareto values to truncation categories that do not include 'Input.'

Of interest is how the bottom four values are all truncation categories with 1-digit truncations including 'Input', and the next four categories are 2-digit truncation including 'Input' rather than the other 1-digit truncations without 'Input' as a truncation category. This indicates that the truncation category does differentiate solution set distance to superfront, suggesting that truncation categories including 'Input' are of lower optimal quality.

Table 4.5 (organized likewise to Table 4.4), also shows a divergence in distances from the superfront between truncation categories containing 'Input' and those that do not. Distance from the superfront (when large) indicates a lack of solution diversity, potentially accounting for the different ordering of truncation categories within each group of decimal truncation from Table 4.4.

Despite noting these trends, all categories have close distance values except for the last four ('Input, Penalty, Constraint', 'Input', 'Input, Constraint', and 'Input, Penalty').

Both Tables 4.4 and 4.5 present sensible results. The consequences of truncating 'Input' (the specific ship dimension values) are much greater compared to truncation of 'Penalty' (the penalty function calculation) and 'Constraint' (the constraint equations) due to the compounding and propagation of error. 'Input' values being truncated have a greater effect as those values are the baseline of all of the optimizer calculations. 'Constraint' and 'Penalty' values are more meta-evaluative, by respectively computing a measure of unfitness and weighing the fitness score. Truncation in these areas does not have nearly the same effect proliferation.

The difference between truncations containing 'Input' and those without is seen for all four of the cases examined in Section 4.4.

Truncation Categories	Digits Truncated	Distance From Superfront (Pareto)	Scaled (%)
No Truncation	**X**	**0.05305**	**0**
Penalty, Constraint	4	0.05372	0.65
Penalty	2	0.05398	0.91
Constraint	4	0.05497	1.87
Input, Constraint	4	0.05581	2.69
Penalty, Constraint	2	0.05612	2.99
Input, Penalty	4	0.05634	3.2
Input, Penalty, Constraint	4	0.05649	3.35
Input	4	0.05666	3.51
Penalty, Constraint	3	0.05691	3.76
Input, Penalty, Constraint	3	0.05714	3.99
Constraint	2	0.05739	4.23
Penalty	4	0.05745	4.29
Penalty	3	0.05788	4.7
Input, Penalty	3	0.05789	4.72
Input	3	0.05867	5.47
Constraint	3	0.05891	5.7
Input, Constraint	3	0.05959	6.37
Constraint	1	0.06282	9.51
Penalty	1	0.06321	9.9
Penalty, Constraint	1	0.0651	11.74
Input	2	0.06996	16.47
Input, Constraint	2	0.07337	19.79
Input, Penalty, Constraint	2	0.07471	21.1
Input, Penalty	2	0.07549	21.85
Input, Penalty	1	0.14201	86.65
Input, Penalty, Constraint	1	0.14355	88.15
Input	1	0.1455	90.04
Input, Constraint	1	0.15572	100

Table 4.5: Average Distance From Superfront (Pareto) for All Datasets

4.3 Description of Cases

Four cases are examined in this dissertation. Section 4.1 describes how the designer determines which terms to consider. Each of the terms (‘more truncation’, ‘useful optimizer’, and ‘enough solutions’) require a quantitative definition.

Possible definitions for ‘more truncation’ include a certain amount of rounding to specific decimal places. Definitions for ‘useful optimizer’ may involve the fraction of total solutions on the final Pareto front, or the number of solutions ever on the Pareto front. ‘Enough solutions’ may be defined as the quantity of feasible solutions, or as a ratio of Pareto solutions to feasible solutions. Any of these definitions can be used.

The designer selects ones she deems representative of her hunch. If she is primarily concerned about how truncation affects feasible solutions, she may define 'enough solutions' as some function of total feasible solutions rather than from the number of Pareto optimal solutions. The corresponding threshold value is also determined from designer perspective.

For all cases presented, Table 4.6 lists threshold values and definitions for each term.

Term	Definition	Threshold
More truncation	component value decimal places	< 3
Useful optimizer	% Pareto final / total solutions	> 0.07696
Enough solutions	% feasible / total solutions	> 0.4778

Table 4.6: Term Definitions and Thresholds

'More truncation' is defined as component values rounded to less than three decimal places. 'Useful optimizer' is defined as percentage of Pareto final solutions to total solutions greater than 0.07696. 'Enough solutions' is defined as percentage of feasible solutions to total solutions greater than 0.4778. These threshold numbers were determined from average values of definitions which attempt to quantify the designer's thoughts behind each term.

These definitions and thresholds may only be a first, initial determination. Further examination and testing may reveal more appropriate values for adjusted definitions, which is not something included in the scope of this work's cases. Using a mean value for initial quantitative thresholds allows for easier identification and thus adjustment of values as needed.

In order to present a manageable case comparison for framework proof of concept, all cases use the same terms and definitions from Table 4.6. There are six possible combinations of syllogism term assignments for the subject, predicate, and middle terms as cases. Of these six, four are selected as relevant conclusions constructed for decision-making.

The middle term is the connecting term between premises. The subject term arbitrates what the conclusion statement is about. The predicate term expresses the essential thought about the subject as a property the subject has or is characterized by.

Due to the designer's concerns about truncation as affecting tool appropriateness, no cases are assigned 'more truncation' as the middle term. The designer certainly wishes to know either about the truncation (subject) or about the subject's truncation (predicate). Should truncation be the connecting (middle) term, the conclusion examined for sufficiency would not be about or determined by truncation and thus of little interest to the designer.

Therefore this chapter will examine four cases out of the six possible combinations. Table 4.7 lists the assignment of the syllogism subject, predicate, and middle terms for each case examined in this chapter.

	Subject Term	Predicate Term	Middle Term
Case 1	More Truncation	Useful Optimizer	Enough Solutions
Case 2	More Truncation	Enough Solutions	Useful Optimizer
Case 3	Enough Solutions	More Truncation	Useful Optimizer
Case 4	Useful Optimizer	More Truncation	Enough Solutions

Table 4.7: Case Term Assignment

All cases will compare the same seven truncation categories (listed in Table 4.1) and use the datasets described in Section 4.2. Each case has a unique term assignment with three syllogism arrangements: deductive (AAA1), abductive (AAA2), and inductive (AAA3) syllogisms.

Within a case, deduction and induction have identical major premises. Deduction and abduction have identical minor premises. All syllogisms for a case have identical conclusions. Separate case conclusions can be compared both quantitatively and semantically. The four case conclusions are semantically distinct and correspond to different aspects of the designer's motivating notions.

Case 1

The three hypothesis syllogisms for Case 1 are as follows:

Deduction (AAA1):

Enough solutions indicate a useful optimizer

A more truncated optimizer produces enough solutions

A more truncated optimizer is still a useful optimizer

Abduction (AAA2):

A useful optimizer produces enough solutions

A more truncated optimizer produces enough solutions

A more truncated optimizer is still a useful optimizer

Induction (AAA3):

Enough solutions indicate a useful optimizer

Enough solutions are generated from more truncated optimizers

A more truncated optimizer is still a useful optimizer

The conclusion statement of Case 1 can be clearly seen to come directly from the designer's hunch that more truncation still allows for a useful optimizer.

In regards to the comparison of two statements with flipped first and second terms (such as the middle term and predicate term of the major premise for deduction and abduction): it can be seen that these two statements cannot be assumed to be the inverse of each other (mutually exclusive). 'Enough solutions indicates a useful optimizer' but not enough solutions does not also indicate if an optimizer is useful or not useful.

Case 2

The three hypothesis syllogisms for Case 2 are as follows:

Deduction (AAA1):

A useful optimizer produces enough solutions

A more truncated optimizer is a useful optimizer

A more truncated optimizer has enough solutions

Abduction (AAA2):

Enough solutions indicate a useful optimizer

A more truncated optimizer is a useful optimizer

A more truncated optimizer has enough solutions

Induction (AAA3):

A useful optimizer produces enough solutions

A useful optimizer can be a more truncated optimizer

A more truncated optimizer has enough solutions

Case 1's conclusion: 'a more truncated optimizer is still a useful optimizer' and Case 2's conclusion: 'a more truncated optimizer has enough solutions' have the same subject term of 'more truncation' with differing predicate terms.

Both conclusions' predicate terms speak to the concerns of the designer. Case 2 follows from a corresponding hunch to of Case 1's motivating questions. Case 1's conclusion came from the designer's hunch that more truncation would still allow for a useful optimizer. A useful optimizer stems from concern over rareness of Pareto solutions from a truncated optimizer. In Case 2, enough solutions is a re-prioritized objective stemming from concern over lack of feasible solutions and inspires this case's conclusion. The predicate term indicates the specifics of the designer's motivations while the subject term provides the context.

The specific impact on sufficiency of assignment of the same subject term with flipped predicate and middle terms can be investigated when corresponding syllogisms from Case 1 and 2 are compared.

Case 3

The three hypothesis syllogisms for Case 3 are as follows:

Deduction (AAA1):

A useful optimizer can be a more truncated optimizer
Enough solutions indicate a useful optimizer
Enough solutions are generated from more truncated optimizers

Abduction (AAA2):

A more truncated optimizer is a useful optimizer
Enough solutions indicate a useful optimizer
Enough solutions are generated from more truncated optimizers

Induction (AAA3):

A useful optimizer can be a more truncated optimizer
A useful optimizer produces enough solutions
Enough solutions are generated from more truncated optimizers

Case 3 and Case 2 have the same middle term and flipped subject and predicate terms. Subject terms set the context of a statement while predicate terms describe the context of a statement.

For Case 3, the subject term is 'enough solutions' instead of the Case 2 term: 'more truncation'. Rather than considering 'enough solutions' as the essential thought about 'more truncation', now 'enough solutions' is the subject and 'more truncation' is the essential thought about it. Depending on the case, 'enough solutions' or 'more truncation' determines the context the other operates in.

Neither of these constructions is necessarily better as both tell different stories about what is needed to reach sufficiency for designer decision-making. Case 2 indicates more concern on truncation producing enough solutions while Case 3 implies that with enough solutions, truncation will be less crucial.

The specific impact of swapped subject and predicate terms on conclusion sufficiency can be investigated when corresponding syllogisms from Case 2 and 3 are compared.

Case 4

The three hypothesis syllogisms for Case 4 are as follows:

Deduction (AAA1):

Enough solutions are generated from more truncated optimizers

A useful optimizer produces enough solutions

A useful optimizer can be a more truncated optimizer

Abduction (AAA2):

A more truncated optimizer has enough solutions

A useful optimizer produces enough solutions

A useful optimizer can be a more truncated optimizer

Induction (AAA3):

Enough solutions are generated from more truncated optimizers

Enough solutions indicate a useful optimizer

A useful optimizer can be a more truncated optimizer

Case 4 and Case 3 share the predicate term: 'more truncation'. As described previously, this conclusion implies that with a useful optimizer, truncation will be less crucial. The subject term provides the context for which the predicate term is describing.

The impact of the assignment of the predicate term can be compared with corresponding syllogisms from Case 3 and 4. By swapping the middle and subject terms, the semantic strength of the middle term connecting the minor and major premises can be examined.

4.4 Case Results

4.4.1 Result Analysis Key

A brief discussion for interpretation of result values is required to better understand each case's probability distribution and mutual information values presented in this section.

CPTs

In order to interpret probability results, each statement below provides a key to understanding different values of each conditional probability table.

P(Minor Premise|Major Premise) can be interpreted as "given the cognizance that the major premise is some value, the minor premise being True is/is not likely." This indicates a strong/weak relationship between the two premises.

P(Conclusion Statement|Major Premise) can be interpreted as "given the cognizance that the major premise is some value, the conclusion being True is/is not likely." This indicates a strong/weak contribution from the major premise and predicate term to the conclusion.

P(Conclusion Statement|Minor Premise) can be interpreted as "given the cognizance that the minor premise is some value, the conclusion being True is/is not likely." This indicates a strong/weak contribution from the minor premise and subject term to the conclusion.

These definitions say 'cognizance' rather than a more absolutest declaration as the nodal state is determined by the term definition set by the designer. A designer defined and quantified these terms according to her perspective. These values are thus not objective truths and should not be treated as such.

Sufficient tool-generated data requires some correlation between the conclusion and the major and minor premises. Additionally, major and minor premises should themselves be connected in this case since they are fundamentally related. Other term or premise selections many not require a correlation between the major and minor premise should they manifest as separate contributions to the conclusion.

Flat Mutual Information

Mutual information is more complicated to interpret than conditional probability. Flat mutual information indicates the amount of mutual dependence between the major premise, minor premise, and the conclusion. It measures how much knowing about any statement, such as the major premise or the minor premise, reduces uncertainty about another statement, such as the conclusion. For example, if the major premise, minor premise, and conclusion are independent, then knowing one does not give any information about the others, so their mutual information is zero. At the other extreme, if one statement is a deterministic function of the the others, then all information conveyed by one is shared with the others. Therefore knowing one statement determines the value of the other statements and vice versa.

For sufficient conclusions for decision-making, a relationship between statements is required as fully independent statements imply unrelated designer inferences. However, fully dependent or overly deterministic relationships are not desirable. Large flat mutual information values indicate duplication of knowledge in the statements and suggests distinct, additional data should be generated and appraised. Deductive reasoning instances structured in logically valid ways may have generated data sufficiency with lower flat mutual information values than induction or abduction, which often require larger values to connect inferences that do not have the benefit of such a structure.

For flat mutual information, a negative value is a sign of synergy between statements; that is, some kind of unexplained amplification related to hidden interdependencies that this framework is not designed to identify. Interdependencies may contribute to emergent design failures that a decision maker wishes to avoid.

Therefore, extreme values for flat mutual information (both very large and very small) should be viewed with skepticism in regards to their merits for decision-making. There is not a numerically ideal number to be reached for sufficient constructions for decision-making as flat mutual information values must be compared across syllogism constructions. Positive flat mutual information values indicate one (or more) of the statements can provide information reducing uncertainty about the others. A zero flat mutual information value means all three statements are independent and thus dubious for decision-making.

Due to the construction of flat mutual information, AAA1 and AAA3 syllogisms have the same value for corresponding truncation categories. This is due to the shared major premise and priority over the minor premise in determining relevancy for decision-making. In order to differentiate between deductive and inductive syllogisms, this work utilizes a layered network with associated mutual information.

Layered Mutual Information

Layered mutual information allows for the differentiation between deductive and inductive syllogisms. Ordering and prioritization of premises imitate the formation of path dependencies in design. By isolating the premises to a combined node, layered mutual information can incisively evaluate the correlation of premises to conclusions without the noise of major premise to minor premise contribution.

Layered mutual information is calculated from a network with a combined node consisting of the major and minor premise. This allows for an evaluation of the quantified mutual dependence of the premises with the conclusion. Layered mutual information provides a more focused examination on how premises together can reduce uncertainty about the conclusion. The structure of layered networks allows for the incorporation of more than two premise nodes if desired, which is not demonstrated in this work.

It is critical that there is a deterministic relationship between the combined premises and the conclusion. Large positive values are desirable as that indicates the conclusion can be understood based off of premise data. Preferably, a conclusion has a strong relationship with the premises it is based on. Deduction, induction, and abduction all have different expectations of combined premise contributions to the conclusion. Deduction, with primarily logically valid structures, requires less correlation to be sufficient. Induction and abduction do not have the benefit of such a structure and so need a larger mutual information value to feel similarity comfortable.

For layered mutual information, a negative value indicates more non-relevant events in comparison to relevant events and is a sign that term definitions should be adjusted. Zero mutual information indicates independence between the premises and the conclusion and is not desirable for decision-making.

4.4.2 Case 1

To enable the evaluation of tool-generated data as design inferences for decision-making, probability distributions and mutual information of the associated network for Case 1 are calculated. They are presented here.

CPTs

Deduction and induction (AAA1 and AAA3) have the same major premise due to their syllogism construction: 'enough solutions indicate a useful optimizer'. The major premise for abduction (AAA2) is 'a useful optimizers produces enough solutions'.

	Major = True		Major = False		Major = N/A	
	AAA1 & AAA3	AAA2	AAA1 & AAA3	AAA2	AAA1 & AAA3	AAA2
Input Penalty Constraint	0.529	0.529	0.442	0.027	0.029	0.444
Input Constraint	0.531	0.531	0.429	0.038	0.04	0.431
Input	0.525	0.525	0.383	0.04	0.092	0.435
Input Penalty	0.54	0.54	0.423	0.031	0.038	0.429
Penalty	0.95	0.95	0.008	0.042	0.042	0.008
Penalty Constraint	0.965	0.965	0.008	0.027	0.027	0.008
Constraint	0.96	0.96	0.006	0.033	0.033	0.006

Table 4.8: Major Premise Probabilities: Case 1

Table 4.8 shows differentiation in probabilities on truncation categories containing 'Input'. When the nodal status of the major premise is True, the differentiation occurs for all three syllogism constructions. When the nodal status of the major premise is False or N/A, the differentiation only appears according to 'useful optimizer' term positioning. For AAA2, 'useful optimizer' determines relevancy as the first term of the major premise and so differentiation in truncation category probability is visible for AAA2 when major premise nodal state is N/A. As the second term for AAA1 and AAA3, it determines if the nodal state is True or False and can be seen for AAA1 and AAA3 when major premise nodal status is False.

	Minor = True		Minor = False		Minor = N/A	
	AAA1 & AAA2	AAA3	AAA1 & AAA2	AAA3	AAA1 & AAA2	AAA3
Input Penalty Constraint	0.494	0.494	0.006	0.477	0.5	0.029
Input Constraint	0.496	0.496	0.004	0.465	0.5	0.04
Input	0.442	0.442	0.058	0.467	0.5	0.092
Input Penalty	0.488	0.488	0.013	0.475	0.5	0.038
Penalty	0.494	0.494	0.006	0.465	0.5	0.042
Penalty Constraint	0.498	0.498	0.002	0.475	0.5	0.027
Constraint	0.496	0.496	0.004	0.471	0.5	0.033

Table 4.9: Minor Statement Probabilities: Case 1

Deduction and abduction (AAA1 and AAA2) have the same minor premise due to their syllogism construction: 'a more truncated optimizer produces enough solutions'.

The major premise for induction (AAA3) is ‘enough solutions are generated from more truncated optimizers’.

Table 4.9 does not show large difference in probability values corresponding to truncation categories. ‘Input’ is the largest value for AAA1 and AAA2 syllogisms with Minor status of False, but the value is still within 0.05 of other AAA1 and AAA2 Minor status False values. The minor premise is associated with the subject and middle terms, and term assignment for Case 1 does not include ‘useful optimizer’ among those.

It should be noted that there appears to be a trend of additive truncation categories; that is ‘Input Penalty’ values are larger than individual ‘Input’ or ‘Penalty’, and ‘Input Penalty Constraint’ is the largest. This trend does not persist completely throughout all cases but in general, truncated categories stack probability values.

	Conclusion = True	Conclusion = False	Conclusion = N/A
Input Penalty Constraint	0.06	0.44	0.5
Input Constraint	0.069	0.431	0.5
Input	0.069	0.431	0.5
Input Penalty	0.075	0.425	0.5
Penalty	0.494	0.006	0.5
Penalty Constraint	0.494	0.006	0.5
Constraint	0.496	0.004	0.5

Table 4.10: Conclusion Statement Probabilities: Case 1

Table 4.10 has the same values for each nodal state for all syllogisms as all syllogisms in a case have the same conclusion: ‘a more truncated optimizer is still a useful optimizer’. Since ‘useful optimizer’ is the predicate term for Case 1, the probability values for conclusion statement nodal states of True and False differentiate based off of truncation categories containing ‘Input’. The subject term ‘more truncation’ determines relevancy. Due to the method of generating evidence (where even amounts of 1,2,3 and 4-digit truncation occurred for all dataset runs) all probability values for conclusion statement N/A nodal status are 0.5.

Probability values of conclusion statements with a True nodal status for truncation categories with 'Input' and those without are distinct. Those containing 'Input' are less than 0.08 while those not containing 'Input' are greater than 0.49. This trend is flipped for probabilities of conclusion statements with False nodal status, where truncation categories containing 'Input' are all greater than 0.42 and those not containing 'Input' are less than 0.01. It is desirable to have a larger True conclusion nodal status as that indicates the definitions chosen to conduct this analysis do not immediately preclude decision construction from the minor and major premises.

It is concerning for decision sufficiency that truncation categories containing 'Input' have such a low True conclusion nodal status. If all truncation categories had similarly low probabilities, it would be recommended to adjust term definitions and reconsider term selection. However, since truncation categories not containing 'Input' have reasonably high True conclusion nodal status probabilities, it is an indication that those truncation categories containing 'Input' are not suitably generated data for decision-making.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0	0.894	0.991	0	0.009	0	0.214	0.786
Input Constraint	0.125	0	0.875	1	0	0	0	0.105	0.895
Input	0.119	0	0.881	0.989	0	0.011	0	**0.636**	**0.364**
Input Penalty	0.127	0	0.873	0.99	0	0.01	0	0.333	0.667
Penalty	0.513	0	0.487	0.75	0	0.25	0	0.15	0.85
Penalty Constraint	0.51	0	0.49	0.75	0	0.25	0	0.077	0.923
Constraint	0.512	0	0.488	**0.667**	0	**0.333**	0	0.125	0.875

Table 4.11: Minor Given Major Statement Probabilities for AAA1 Syllogism: Case 1

Table 4.11 shows the probabilities of the minor premise given the major premise of deduction (AAA1). As expected, differentiation on truncation categories containing 'Input' occurs for minor premise given major premise nodal states of True and False, and not for N/A. This differentiation is much more pronounced for minor premise given major premise True than for minor premise given major premise False.

The probability values of zero for minor premise False nodal status given major premise nodal states of True or False and minor premise True status given major premise N/A status are due to the syllogism structure.

The bolded values are interesting. ‘Input’ has a much larger False minor premise nodal status probability given N/A major premise nodal status, and also a corresponding small N/A minor premise nodal status probability given N/A major premise nodal status.

Of truncation categories without ‘Input’, there may be an indication of between ‘Penalty’ and ‘Constraint’ which one is more sufficient for decision-making. ‘Constraint’ truncation’s probability value for minor premise N/A nodal status given major premise False nodal status is 0.333, which is significantly larger in comparison to other truncation categories not containing ‘Input’. The same trend is apparent for minor premise True nodal status given major premise False status, where ‘Constraint’ truncation category has significantly smaller probability values than other truncation categories without ‘Input’.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0	0.894	0	0.154	0.846	0.986	0.005	0.009
Input Constraint	0.125	0	0.875	0	0.056	0.944	0.995	0.005	0
Input	0.119	0	0.881	0	0.158	0.842	**0.871**	**0.12**	0.01
Input Penalty	0.127	0	0.873	0	0.2	0.8	0.976	0.015	0.01
Penalty	0.513	0	0.487	0	0.15	0.85	0.75	**0**	0.25
Penalty Constraint	0.51	0	0.49	0	0.077	0.923	0.75	**0**	0.25
Constraint	0.512	0	0.488	0	0.125	0.875	**0.667**	**0**	**0.333**

Table 4.12: Minor Given Major Statement Probabilities for AAA2 Syllogism: Case 1

Table 4.12 shows the probabilities of the minor premise given the major premise of abduction (AAA2). As expected, differentiation on truncation categories containing ‘Input’ occurs for minor premise given major premise nodal states of True and N/A, and not for False major premise nodal status. This differentiation is much more pronounced for minor premise given major premise True than for minor premise given major premise N/A.

The probability values of zero for minor premise False nodal status given major premise True and minor premise True status given major premise False are due to the syllogism structure. If the major premise has a True nodal status, then the minor premise nodal state must be either True or N/A (it cannot be False). If the major premise has a False nodal status, the minor premise nodal state must be False or N/A (it cannot be True). The structure of the syllogism requires this as AAA2 has the middle term as the second term in both the minor and major premises.

The bolded values are curious. 'Input' has a much larger probability for minor premise False nodal status given major premise N/A nodal status, and also a corresponding relatively small minor premise True nodal status probability given major premise N/A status when compared to other truncations categories containing 'Input'. For truncations without 'Input', when the minor premise nodal status is False given a N/A major premise nodal status, the probability is zero. This in contrast with truncations containing 'Input' for the same circumstances that have small but existing probabilities implies some not understood effect of truncation categories.

Similar to AAA1 in Table 4.11, there may be an indication between 'Penalty' and 'Constraint' of which one is more sufficient for decision-making. The probability values of truncation category 'Constraint' for minor premise N/A nodal status given major premise N/A nodal status is 0.333, which is significantly larger in comparison to other truncation categories not containing 'Input'. Because of the zero probability value of 'Constraint' for minor premise False nodal status given major premise N/A status, the same trend is apparent for minor premise True given major premise N/A.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0.894	0	0.991	0.009	0	0	0	1
Input Constraint	0.125	0.875	0	1	0	0	0	0	1
Input	0.119	0.881	0	0.989	0.011	0	0	0	1
Input Penalty	0.127	0.873	0	0.99	0.01	0	0	0	1
Penalty	0.513	0.487	0	0.75	0.25	0	0	0	1
Penalty Constraint	0.51	0.49	0	0.75	0.25	0	0	0	1
Constraint	0.512	0.488	0	**0.667**	**0.333**	0	0	0	1

Table 4.13: Minor Given Major Statement Probabilities for AAA3 Syllogism: Case 1

Table 4.13 shows the probabilities of the minor premise given the major premise of induction (AAA3). As expected, differentiation on truncation categories containing 'Input' occurs for minor premise given major premise nodal states of True and False, and not for major premise N/A nodal status.

The syllogism structure of induction has the middle term as the first term in both the minor and major premises, therefore making it so that the minor premise has N/A nodal status if and only if the major premise nodal status is N/A.

There is a larger difference between probabilities for truncation categories that contain and do not contain 'Input' for minor premise given major premise True and False nodal states (0.4 compared to 0.2).

For all syllogism constructions of deduction, abduction, and induction, truncations without 'Input' have probability values for minor premise True nodal status given major premise True nodal status just over 0.5. These circumstances indicate possible sufficiency of the premises. The designer wants the minor premise to follow from the major premise, and a higher probability indicates that. It is a good sign of sufficiency for a conclusion to be constructed from two statements with True nodal status, though a caveat for this situation is statements are primarily True only for truncation categories not including 'Input'.

Despite their True nodal status, these probabilities do not indicate actual correctness but rather affirm the designer's construction of what those premises are defined as.

Similar to AAA1 in Table 4.11 and 4.12, the 'Constraint' category truncation probability value for minor premise False nodal status given major premise False nodal status is 0.333, which is significantly larger in comparison to other truncation categories not containing 'Input'. Because of the zero probability value of 'Constraint' category truncation for N/A minor premise given False major premise, the same trend is apparent for True minor premise given False major premise.

Since deduction (AAA1) and induction (AAA3) have identical major premises

	Conclusion |Major = True			Conclusion |Major = False			Conclusion |Major = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0	0.894	0	0.991	0.009	0.143	0.071	0.786
Input Constraint	0.125	0	0.875	0	1	0	0.053	0.053	0.895
Input	0.119	0	0.881	0	0.989	0.011	0.068	**0.568**	0.364
Input Penalty	0.127	0	0.873	0	0.99	0.01	0.167	0.167	0.667
Penalty	0.513	0	0.487	0	0.75	0.25	0.15	0	0.85
Penalty Constraint	0.51	0	0.49	0	0.75	0.25	0.077	0	0.923
Constraint	0.512	0	0.488	0	**0.667**	**0.333**	0.125	0	0.875

Table 4.14: Conclusion Given Major Statement Probabilities for All Truncations of AAA1 and AAA3 Syllogisms: Case 1

and conclusions, Table 4.14 shows conclusion probabilities given major premise for both AAA1 and AAA3. Syllogism structure also requires that conclusion statements with nodal status False have zero probability given major premise status of True, and also for conclusion True nodal status for major premise False nodal status.

If the major premise has a N/A nodal status, 'Input' truncation has a 0.568 probability of a False nodal status, significantly more than any other False conclusion status given N/A major status.

Probability values for 'Constraint' truncation category False and N/A conclusion nodal state given False major premise state are similar to what was seen throughout all three syllogisms' minor premise given major premise tables. This could indicate 'Constraint' being more sufficient for decision-making over 'Penalty' but is not conclusive.

	Conclusion |Major = True			Conclusion |Major = False			Conclusion |Major = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0	0.894	0.154	0	0.846	0	0.991	0.009
Input Constraint	0.125	0	0.875	0.056	0	0.944	0	1	0
Input	0.119	0	0.881	0.158	0	0.842	0	0.99	0.01
Input Penalty	0.127	0	0.873	0.2	0	0.8	0	0.99	0.01
Penalty	0.513	0	0.487	0.15	0	0.85	0	0.75	0.25
Penalty Constraint	0.51	0	0.49	0.077	0	0.923	0	0.75	0.25
Constraint	0.512	0	0.488	0.125	0	0.875	0	**0.667**	**0.333**

Table 4.15: Conclusion Given Major Statement Probabilities for AAA2 Syllogism: Case 1

Table 4.15 shows conclusion probabilities given major premise for abduction (AAA2). It has the expected differentiation along 'Input' truncation categories for conclusion given major premise True and N/A nodal states. Conclusion status cannot be False if the major premise is relevant (does not have a N/A nodal status) and cannot be True if the major premise has a N/A nodal status. This is due to the AAA2 syllogism structure where the predicate term is the first term of the major premise and the second term of the conclusion.

The same trend in regards to 'Constraint' truncation category conclusion nodal states of False and N/A given N/A major premise state (bolded) is seen.

	Conclusion |Minor = True			Conclusion |Minor = False			Conclusion |Minor = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.114	0.886	0	0.667	0.333	0	0	0	1
Input Constraint	0.134	0.866	0	0.5	0.5	0	0	0	1
Input	0.142	0.858	0	**0.107**	**0.893**	0	0	0	1
Input Penalty	0.141	0.859	0	0.5	0.5	0	0	0	1
Penalty	0.987	0.013	0	1	0	0	0	0	1
Penalty Constraint	0.987	0.013	0	1	0	0	0	0	1
Constraint	0.992	0.008	0	1	0	0	0	0	1

Table 4.16: Conclusion Given Minor Statement Probabilities for AAA1 and AAA2 Syllogisms: Case 1

Since deduction (AAA1) and abduction (AAA2) have identical minor premises and conclusions, Table 4.16 shows conclusion probabilities given minor premise for both AAA1 and AAA2. Syllogism structure also requires that conclusion statements with nodal status N/A have zero probability given minor premise states of True or False. AAA1 and AAA2 have the subject term as the first term in both the minor premise and the conclusion, which manifests as probability of zero for conclusion nodal states True and False given minor premise N/A nodal status and a probability of one for conclusion N/A status given minor premise N/A status. The minor premise determines the relevance of the conclusion statement for deduction and abduction.

As expected, there is a differentiation between truncation categories that include and exclude 'Input' for conclusion given minor states True or False. 'Input' category differentiates due to conclusion relevance.

The bolded values of 'Input' truncation category are significantly different than other truncations for conclusion states True and False given minor premise False nodal status. The probability value for conclusion status True given minor premise False nodal status is 0.107 which is less than a quarter of the next smallest value. This is a strong indication that 'Input' truncation compared to 'Penalty' and 'Constraint' truncations are vastly different in terms of sufficiency. Potentially, these categories are so opposite that the combination truncations (such as 'Input Penalty' and 'Input Constraint') seem to split the difference in probability values. 'Input Penalty Constraint' is the closest combination truncation in value to the truncation categories not containing 'Input'. This may indicate that the probability or circumstances surrounding the truncations are not fully additive after all if 'Penalty' and 'Constraint' dilute 'Input' truncation category's affect on the probability value.

It is especially interesting because the desirable probability for conclusion status True given minor premise status False is a low value. There is a reversal here for truncations containing 'Input' and those not. Possibly, this is due to the small amount of evidence for False minor premise nodal status for AAA1 and AAA2 as seen in Table 4.9.

	Conclusion |Minor = True			Conclusion |Minor = False			Conclusion |Minor = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.114	0.886	0	0	0	1	0.143	0.071	0.786
Input Constraint	0.134	0.866	0	0	0	1	0.053	0.053	0.895
Input	0.142	0.858	0	0	0	1	0.068	**0.568**	**0.364**
Input Penalty	0.141	0.859	0	0	0	1	0.167	0.167	0.667
Penalty	0.987	0.013	0	0	0	1	0.15	0	0.85
Penalty Constraint	0.987	0.013	0	0	0	1	0.077	0	0.923
Constraint	0.992	0.008	0	0	0	1	0.125	0	0.875

Table 4.17: Conclusion Given Minor Statement Probabilities for AAA3 Syllogism: Case 1

Table 4.17 shows conclusion probabilities given minor premise for induction (AAA3). Syllogism structure of AAA3 requires that conclusion with nodal status N/A have zero probability given minor premise status of True. AAA3 has the subject term positioned as the second term of the minor premise and the first term of the conclusion, which manifests as zero probabilities for conclusion nodal states True and False given minor premise False nodal status and a probability of one for conclusion N/A status given minor premise False.

As expected, there is a difference in probability values between truncation categories including ‘Input’ for conclusion given minor status True and to some extent False.

Truncation categories not containing ‘Input’ have much higher probability values for conclusion statement True nodal status given minor premise True nodal status (over 0.98 compared to under 0.15). This is a strong indication of induction’s ability to form sufficient decisions based off of the minor premise for truncation categories not containing ‘Input’.

The bolded values of ‘Input’ truncation are significantly different than other truncations for conclusion states False and N/A given minor premise N/A nodal status. The probability value for conclusion status False given minor premise N/A nodal status is 0.568 which is quadruple the next largest value. The probability value for conclusion status N/A given minor premise N/A nodal status is 0.364 which is half the next smallest value. Unlike Table 4.16 where a similar trend comes from the small amount of evidence for False minor premise nodal states for AAA1 and AAA2, Table 4.9 indicates that lack of evidence is not the reason behind Table 4.17’s trend. More investigation is required to understand this occurrence.

Flat Mutual Information

Flat mutual information represents the amount of mutual dependence between the three distinct statements of the major premise, the minor premise, and the conclusion statement.

Figure 4.1 shows large AAA1 and AAA3 values for truncation categories containing ‘Input’ (between 0.169 and 0.212) and small AAA1, AAA2, and AAA3 values for truncation categories not containing ‘Input’ (between 0.005 and 0.015). Interestingly, AAA2 mutual information values for truncation containing ‘Input’ are negative

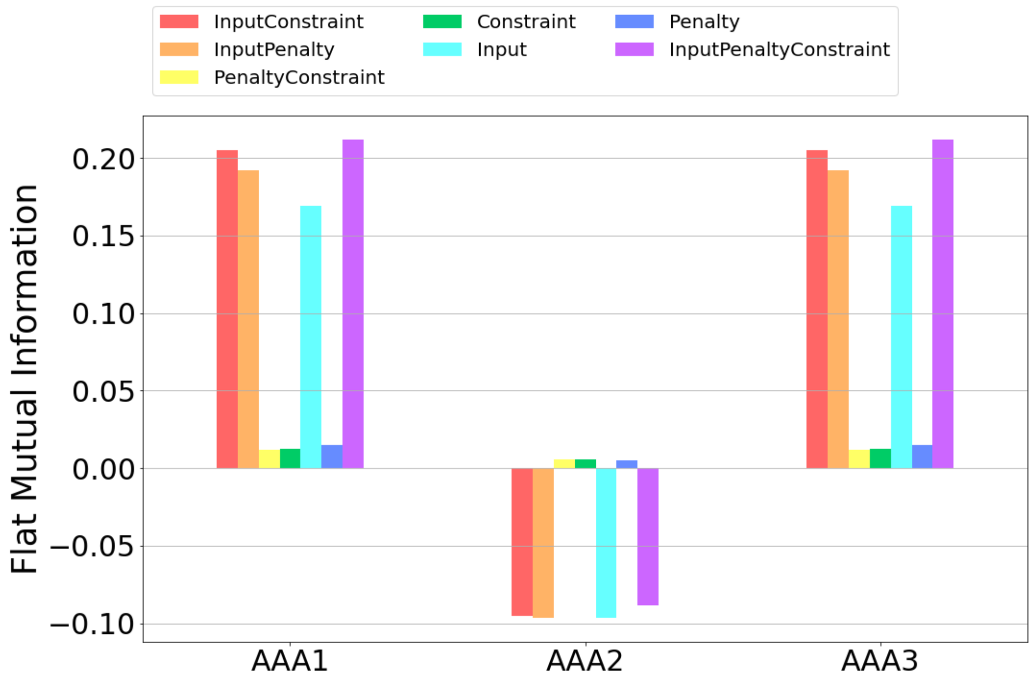

Figure 4.1: Flat Mutual Information: Case 1

	AAA1	AAA2	AAA3
Input Constraint	0.205	-0.096	0.205
Input Penalty	0.192	-0.096	0.192
Penalty Constraint	0.012	0.006	0.012
Constraint	0.013	0.006	0.013
Input	0.169	-0.097	0.169
Penalty	0.015	0.005	0.015
Input Penalty Constraint	0.212	-0.088	0.212

Table 4.18: Flat Mutual Information Values for Case 1

(between -0.088 and -0.096).

Abduction, in comparison to deduction and induction, has a less rigorous structure and thus less defined connections between statements. Abduction is an articulation of an insight or hunch rather than a more direct application of a rule (deduction) or test (induction), and so requires some amount of creativity. Negative values here may indicate some emergent relationship between and due to statements rather than an expected reduction in uncertainty had the mutual information values been positive. Only truncation categories containing ‘Input’ are negative.

Within truncation categories containing ‘Input’, visible differences in mutual information values imply the additive nature of truncation categories. For AAA1 and AAA3, ‘Input’ alone is the smallest value and ‘Input, Penalty, Constraint’ is the largest with ‘Input, Penalty’ and ‘Input, Constraint’ in the middle. The same trend is present for AAA2 but the values are closer together as seen in Table 4.18.

It is apparent from the previously discussed probability distributions that truncations containing ‘Input’ are not likely to be sufficient for decision-making. Since ‘Input’, ‘Penalty’, and ‘Constraint’ are representative of their combinatoric trends, flat mutual information values for them are compared to investigate this in Table 4.19.

Syllogism	Constraint	Penalty	Input
AAA1 and AAA3	0.013	0.015	0.169
AAA2	0.006	0.005	-0.097

Table 4.19: Comparing Flat Mutual Information Values for Singular Truncation Categories

‘Constraint’ and ‘Penalty’ are much smaller than ‘Input’. ‘Constraint’ is slightly smaller than ‘Penalty’ for AAA1 and AAA3, but slightly larger for AAA2. Due to the magnitude of ‘Input’ values for all three syllogisms, ‘Input’ is clearly inappropriate for sufficient decision-making. The negativity of the ‘Input’ value for AAA2 is not why it is not sufficient, but rather the largeness of the value is. Layered mutual information will provide insights to differentiate between ‘Penalty’ and ‘Constraint’ and AAA1 and AAA3.

Layered Mutual Information

As mentioned previously, flat mutual information does not distinguish between AAA1 and AAA3 syllogisms. Layered mutual information provides a differentiation between deduction and induction and more clearly identifies the relationship between the premises and the conclusion statement by combining both premises into one node.

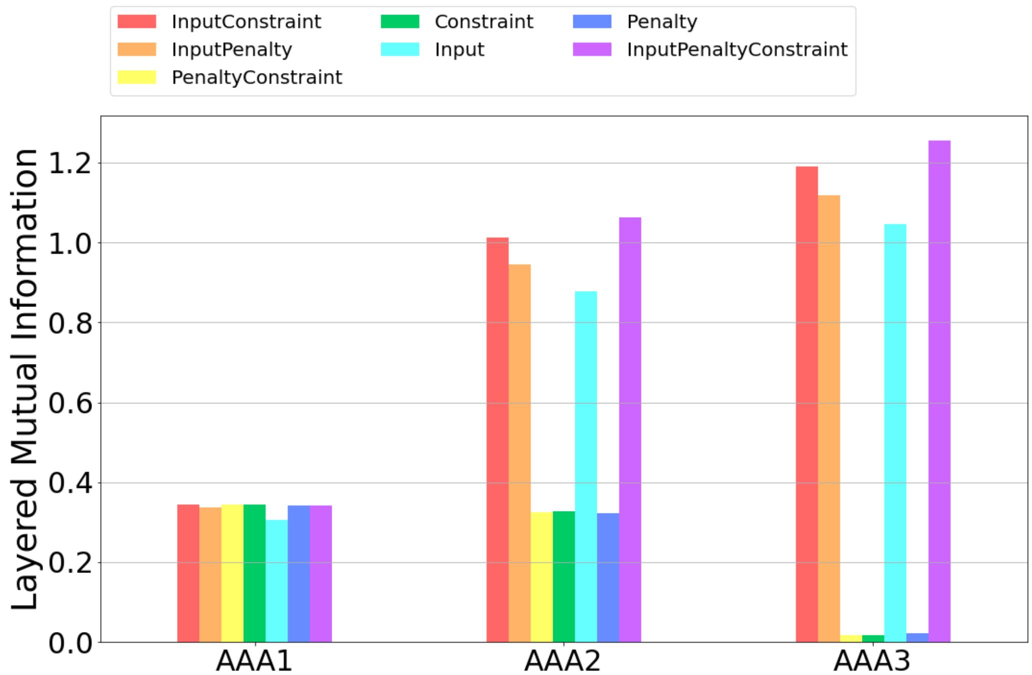

Figure 4.2: Layered Mutual Information: Case 1

	AAA1	AAA2	AAA3
Input Constraint	0.344	1.013	1.189
Input Penalty	0.338	0.945	1.117
Penalty Constraint	0.345	0.324	0.018
Constraint	0.344	0.328	0.017
Input	0.306	0.878	1.045
Penalty	0.342	0.321	0.021
Input Penalty Constraint	0.342	1.063	1.255

Table 4.20: Layered Mutual Information Values: Case 1

Figure 4.2 shows layered mutual information values for Case 1. AAA1 has similar layered mutual information values for all truncation categories. AAA2 has slight separation in values between truncation categories with 'Input' and those without. The separation between truncation categories is more pronounced for AAA3. Trivially it would be understandable for AAA2 and AAA1 to have the most disparate values with AAA3 somewhere in the middle as a general trend of structure strength, but that does not agree with what these results show. This suggests that the framework developed is able to make implications beyond the construction and ordering of terms within the premises. Syllogism structure is only one part of what makes up a sufficient decision, and this shows that designer choice and conclusion affect generated solution data.

Layered mutual information values suggest that when structured as a deductive syllogism, any truncation category has a deterministic relationship between premises and conclusion. It also implies the same for abduction truncation categories not including 'Input'. Induction has (corresponding to truncation categories with and without 'Input') both too much and not enough dependence between the premises and the conclusion for sufficient decision-making. However, sufficiency cannot be definitively stated yet without comparative examinations of other premise constructions. Other conclusions or combinations of terms may provide additional insights to extend or narrow these judgements.

Case 1 demonstrates the value of the syllogistic Bayesian framework. Conditional probability, flat mutual information, and layered mutual information values are calculated to evaluate the generated data for design knowledge queries. Cases 2-4 will show additional syllogism constructions for comparable evaluation of different term assignments.

4.4.3 Case 2

To enable the evaluation of generated tool data as design inferences for decision-making, probability distributions and mutual information of the associated network for Case 2 are calculated. They are presented here.

CPTs

Due to their syllogism constructions, deduction and induction (AAA1 and AAA3) have the same major premise: 'a useful optimizer produces enough solutions'. The major premise for abduction (AAA2) is 'enough solutions indicate a useful optimizer'.

	Major = True		Major = False		Major = N/A	
Syllogism:	AAA1&AAA3	AAA2	AAA1&AAA3	AAA2	AAA1&AAA3	AAA2
Input Penalty Constraint	0.529	0.529	0.027	0.442	0.444	0.029
Input Constraint	0.531	0.531	0.038	0.429	0.431	0.04
Input	0.525	0.525	0.04	0.383	0.435	0.092
Input Penalty	0.54	0.54	0.031	0.423	0.429	0.038
Penalty	0.95	0.95	0.042	0.008	0.008	0.042
Penalty Constraint	0.965	0.965	0.027	0.008	0.008	0.027
Constraint	0.96	0.96	0.033	0.006	0.006	0.033

Table 4.21: Major Premise Probabilities for All Truncations: Case 2

Table 4.21 presents a difference in probability values for truncation categories containing 'Input' and those not. When the nodal status of the major premise is True, the differentiation occurs for all three syllogism constructions. When the nodal state of the major premise is False or N/A, the differentiation only appears according to 'useful optimizer' term positioning. For AAA1 and AAA3, 'useful optimizer' determines relevancy as the first term of the major premise and so differentiation in truncation category probability is visible for AAA1 and AAA3 when major premise nodal status is N/A. As the second term for AAA2, it determines if nodal state is True or False and can be seen when major premise nodal status is False.

	Minor = True		Minor = False		Minor = N/A	
Syllogism:	AAA1&AAA2	AAA3	AAA1&AAA2	AAA3	AAA1&AAA2	AAA3
Input Penalty Constraint	0.06	0.06	0.44	0.496	0.5	0.444
Input Constraint	0.069	0.069	0.431	0.5	0.5	0.431
Input	0.069	0.069	0.431	0.496	0.5	0.435
Input Penalty	0.075	0.075	0.425	0.496	0.5	0.429
Penalty	0.494	0.494	0.006	0.498	0.5	0.008
Penalty Constraint	0.494	0.494	0.006	0.498	0.5	0.008
Constraint	0.496	0.496	0.004	0.498	0.5	0.006

Table 4.22: Minor Statement Probabilities: Case 2

Deduction and abduction (AAA1 and AAA2) have the same minor premise due to their syllogism construction: 'a more truncated optimizer is a useful optimizer'. The major premise for induction (AAA3) is 'a useful optimizer can be a more truncated optimizer'.

Table 4.22 shows differentiation in probabilities on truncation categories containing 'Input'. When the nodal status of the minor premise is True, the differentiation occurs for all three syllogism constructions. When the nodal state of the major premise is False or N/A, the differentiation only appears according to 'useful optimizer' term positioning. For AAA3, 'useful optimizer' determines relevancy as the first term of the minor premise and so differentiation in truncation category probability is visible for AAA3 when major premise nodal status is N/A. As the second term for AAA1 and AAA2, it determines True/False status and can be seen when minor premise nodal status is False.

AAA1 and AAA2 have constant probability values of 0.5 for the minor premise status of N/A due to the minor premise's first term of 'more truncation' which determines relevancy. Due to the method of generating evidence (where even amounts of 1,2,3 and 4-digit truncation occurred for all dataset runs), AAA1 and AAA2 probabilities for minor premise nodal status of N/A are 0.5.

	Conclusion = True	Conclusion = False	Conclusion = N/A
Input Penalty Constraint	0.494	0.006	0.5
Input Constraint	0.496	0.004	0.5
Input	**0.442**	0.058	0.5
Input Penalty	0.488	0.013	0.5
Penalty	0.494	0.006	0.5
Penalty Constraint	0.498	0.002	0.5
Constraint	0.496	0.004	0.5

Table 4.23: Conclusion Statement Probabilities: Case 2

Table 4.23 has the same values for each nodal state for all syllogisms since all syllogisms in a case have the same conclusion: 'a more truncated optimizer has enough solutions'. Since 'useful optimizer' is the middle term for Case 2, there is no differentiation based off of truncation categories containing 'Input' for the conclusion statement as the conclusion is composed of the subject and predicate terms. The subject term is 'more truncation' and determines relevancy. Due to the method of generating evidence (where even amounts of 1,2,3 and 4-digit truncation occurred for all dataset runs), all conclusion statement N/A nodal status probabilities are 0.5.

Probability values for conclusion statements with True nodal status for all truncation categories are all greater than 0.44, with the lowest being the truncation category of 'Input' at 0.442 (bolded), about 0.05 less than most other truncation categories probability values. It is desirable to have a larger True conclusion nodal status as that indicates the definitions chosen to conduct this analysis do not immediately preclude decision construction from the major and minor premises due to the lack of relationship. In this case, all truncation categories have intermediate to high True conclusion nodal status probability values, indicating the connection is strong enough to draw conclusions.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0	0.894	0.154	0	0.846	0	0.991	0.009
Input Constraint	0.125	0	0.875	0.056	0	0.944	0	1	0
Input	0.119	0	0.881	0.158	0	0.842	0	0.99	0.01
Input Penalty	0.127	0	0.873	0.2	0	0.8	0	0.99	0.01
Penalty	0.513	0	0.487	0.15	0	0.85	0	0.75	0.25
Penalty Constraint	0.51	0	0.49	0.077	0	0.923	0	0.75	0.25
Constraint	0.512	0	0.488	0.125	0	0.875	0	0.667	0.333

Table 4.24: Minor Given Major Statement Probabilities for AAA1 Syllogism: Case 2

Table 4.24 shows the probabilities of the minor premise given the major premise of deduction (AAA1). As expected, differentiation of probability values for truncation categories containing 'Input' occurs for minor premise given major premise nodal states of True and N/A, and not for False major premise nodal status, similar to Table 4.21. This differentiation is much more pronounced for minor premise given major premise True status than for minor premise given major premise N/A.

The zero probability values for False minor premise given True or False major premise and True minor premise given N/A major premise are due to the syllogism structure.

Of truncation categories without 'Input', there may be an indication between 'Penalty' and 'Constraint' of which one is more sufficient for decision-making. 'Constraint' truncation category probability value for minor premise N/A nodal status given major premise False nodal status is 0.333, which is significantly larger in comparison to other truncation categories not containing 'Input'. The same trend is apparent for minor premise True given major premise False, where 'Constraint' truncation category probability is significantly less than other truncation categories without 'Input'. These numbers could be due to the lack of evidence when the major premise nodal status is N/A as seen in Table 4.21 or they may indicate some other distinction.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0	0.894	0	0.991	0.009	0.143	0.071	0.786
Input Constraint	0.125	0	0.875	0	1	0	0.053	0.053	0.895
Input	0.119	0	0.881	0	0.989	0.011	0.068	**0.568**	**0.364**
Input Penalty	0.127	0	0.873	0	0.99	0.01	0.167	0.167	0.667
Penalty	0.513	0	0.487	0	0.75	0.25	0.15	**0**	0.85
Penalty Constraint	0.51	0	0.49	0	0.75	0.25	0.077	**0**	0.923
Constraint	0.512	0	0.488	0	**0.667**	**0.333**	0.125	**0**	0.875

Table 4.25: Minor Given Major Statement Probabilities for AAA2 Syllogism: Case 2

Table 4.25 shows the probabilities of the minor premise given the major premise for abduction (AAA2). As expected, differentiation on truncation categories containing 'Input' occurs for minor premise given major premise nodal states of True and False, and not for N/A major premise nodal status. This differentiation is much more pronounced for minor premise given major premise True than for minor premise given major premise N/A.

The probability values of zero for False minor premise given True major premise and True minor premise given False major premise are due to the syllogism structure. If the major premise has a True nodal status, then the minor premise nodal state must be either True or N/A (it cannot be False). If the major premise has a False nodal status, the minor premise nodal state must be False or N/A (it cannot be True). The structure of the syllogism requires this as AAA2 has the middle term as the second term in both the minor and major premises.

The bolded values are interesting. 'Input' has a much larger False minor premise nodal status probability given N/A major premise nodal status probability, and also a corresponding relatively small True minor premise nodal status probability given N/A major premise nodal status probability when compared to other truncations including 'Input'. For truncation categories without 'Input', when the minor premise nodal status is False given a N/A major premise nodal status, the probability is zero. This in contrast with truncation categories containing 'Input' for the same circumstances that have small probabilities implies some not understood effect of truncation categories.

Similar to AAA1 in Table 4.24, there may be an indication between 'Penalty' and 'Constraint' of which one is more sufficient for decision-making. 'Constraint' truncation's probability value for minor premise N/A nodal status given major premise False nodal status is 0.333, which is significantly larger in comparison to other truncation categories not containing 'Input'. Because of the zero probability value of 'Constraint' for False minor premise given N/A major premise, the same trend is apparent for False minor premise given False major premise.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0.894	0	0.154	0.846	0	0	0	1
Input Constraint	0.125	0.875	0	0.056	0.944	0	0	0	1
Input	0.119	0.881	0	0.158	0.842	0	0	0	1
Input Penalty	0.127	0.873	0	0.2	0.8	0	0	0	1
Penalty	0.513	0.487	0	0.15	0.85	0	0	0	1
Penalty Constraint	0.51	0.49	0	0.077	0.923	0	0	0	1
Constraint	0.512	0.488	0	0.125	0.875	0	0	0	1

Table 4.26: Minor Given Major Statement Probabilities for AAA3 Syllogism: Case 2

Table 4.26 shows the probabilities of the minor premise given the major premise of induction (AAA3). As expected, differentiation on truncation categories containing 'Input' occurs for minor premise given major premise nodal status of True, and not for False or N/A major premise nodal states.

The syllogism structure of induction requires the middle term as the first term in both the minor and major premises, thereby making it so that the minor premise has a N/A nodal status if and only if the major premise nodal status is N/A.

For all syllogism constructions of deduction, abduction, and induction, truncations without 'Input' have the probability values for True minor premise nodal status given True major premise nodal status just over 0.5. The designer wants the minor premise to follow from the major premise, and a higher probability indicates this. This is a positive indication of sufficiency for truncation categories not including 'Input'.

	Conclusion |Major = True			Conclusion |Major = False			Conclusion |Major = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0	0.894	0	0.154	0.846	0.986	0.005	0.009
Input Constraint	0.125	0	0.875	0	0.056	0.944	0.995	0.005	0
Input	0.119	0	0.881	0	0.158	0.842	0.871	0.12	0.01
Input Penalty	0.127	0	0.873	0	0.2	0.8	0.976	0.015	0.01
Penalty	0.513	0	0.487	0	0.15	0.85	0.75	0	0.25
Penalty Constraint	0.51	0	0.49	0	0.077	0.923	0.75	0	0.25
Constraint	0.512	0	0.488	0	0.125	0.875	**0.667**	0	**0.333**

Table 4.27: Conclusion Given Major Statement Probabilities for AAA1 and AAA3 Syllogisms: Case 2

Since deduction (AAA1) and induction (AAA3) have identical major premises and conclusions, Table 4.27 shows conclusion probabilities given major premise for both AAA1 and AAA3. Syllogism structure also requires that conclusion statements with nodal status False have zero probability given major premise status of True, and also for conclusion statements True nodal status given major premise False nodal status.

'Constraint' truncation category False and N/A conclusion nodal state given False major premise status has similar probability values to what was seen for AAA1 and AAA2's minor premise given major premise tables, potentially indicating 'Constraint' being more sufficient for decision-making over 'Penalty' but is inconclusive. It is interesting that AAA3 did not exhibit this behavior for minor premise given major premise, but is showing it here for conclusion given major premise.

Table 4.28 shows conclusion probabilities given major premise for abduction (AAA2). It has the expected differentiation along 'Input' truncation categories for conclusion given major premise True and False nodal states. Conclusion status cannot be False if the major premise is relevant (does not have a N/A nodal status) and cannot be True if the major premise has a N/A nodal status. This is due to the AAA2 syllo-

Conclusion Status:	Conclusion |Major = True True	False	N/A	Conclusion |Major = False True	False	N/A	Conclusion |Major = N/A True	False	N/A
Input Penalty Constraint	0.106	0	0.894	0.991	0	0.009	0	0.214	0.786
Input Constraint	0.125	0	0.875	1	0	0	0	0.105	0.895
Input	0.119	0	0.881	0.989	0	0.011	0	**0.636**	**0.364**
Input Penalty	0.127	0	0.873	0.99	0	0.01	0	0.333	0.667
Penalty	0.513	0	0.487	0.75	0	0.25	0	0.15	0.85
Penalty Constraint	0.51	0	0.49	0.75	0	0.25	0	0.077	0.923
Constraint	0.512	0	0.488	**0.667**	0	**0.333**	0	0.125	0.875

Table 4.28: Conclusion Given Major Statement Probabilities for AAA2 Syllogism: Case 2

gism structure where the predicate term is the first term of the major premise and the second term of the conclusion.

If the major premise has a N/A nodal status, 'Input' truncation has a 0.636 probability of a False nodal status, significantly more than any other False conclusion status given N/A major status.

The trend is repeated in regards to 'Constraint' truncation's True and N/A conclusion nodal states given False major premise status (bolded in Table 4.28).

Conclusion Status:	Conclusion |Minor = True True	False	N/A	Conclusion |Minor = False True	False	N/A	Conclusion |Minor = N/A True	False	N/A
Input Penalty Constraint	0.931	0.069	0	0.995	0.005	0	0	0	1
Input Constraint	0.97	0.03	0	0.995	0.005	0	0	0	1
Input	0.909	0.091	0	0.879	0.121	0	0	0	1
Input Penalty	0.917	0.083	0	0.985	0.015	0	0	0	1
Penalty	0.987	0.013	0	1	0	0	0	0	1
Penalty Constraint	0.996	0.004	0	1	0	0	0	0	1
Constraint	0.992	0.008	0	1	0	0	0	0	1

Table 4.29: Conclusion Given Minor Statement Probabilities for AAA1 and AAA2 Syllogisms: Case 2

Deduction (AAA1) and abduction (AAA2) have identical minor premises and conclusions, thus Table 4.29 shows conclusion probabilities given minor premise for both AAA1 and AAA2. Syllogism structure also requires that conclusion with nodal status N/A have zero probability given minor premise states of True or False. AAA1 and AAA2 have the subject term as the first term in both the minor premise and the conclusion statement, which manifests as zero probabilities for conclusion nodal states True and False given minor premise N/A nodal status and a probability of one for conclusion N/A status given minor premise N/A status. The minor premise determines the relevance of the conclusion statement for deduction and abduction.

As expected, there is no differentiation between truncation categories including 'Input'.

The high probabilities (over 0.9) of conclusion True nodal status given minor premise True nodal status are interesting, especially in contrast with the likewise high probabilities of conclusion True nodal status given minor premise False nodal status. In isolation, the first would indicate sufficiency for decision-making, but when compared, it may be a sign only of an appropriately defined conclusion term. More investigation is required to understand this occurrence.

	Conclusion |Minor = True			Conclusion |Minor = False			Conclusion |Minor = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.931	0.069	0	0	0	1	0.986	0.005	0.009
Input Constraint	0.97	0.03	0	0	0	1	0.995	0.005	0
Input	0.909	0.091	0	0	0	1	0.871	0.12	0.01
Input Penalty	0.917	0.083	0	0	0	1	0.976	0.015	0.01
Penalty	0.987	0.013	0	0	0	1	0.75	0	0.25
Penalty Constraint	0.996	0.004	0	0	0	1	0.75	0	0.25
Constraint	0.992	0.008	0	0	0	1	**0.667**	0	**0.333**

Table 4.30: Conclusion Given Minor Statement Probabilities for AAA3 Syllogism: Case 2

Table 4.30 shows conclusion probabilities given minor premise for induction (AAA3). Syllogism structure of AAA3 requires that conclusion statement with nodal status N/A have zero probability given minor premise status of True. AAA3 has the subject term as the second term of the minor premise and the first term of the conclusion, which manifests as zero probabilities for conclusion nodal states True and False given minor premise False nodal status and a probability of one for conclusion N/A status given minor premise False status .

As expected, there is a differentiation between truncation categories including 'Input' for conclusion statements given minor status N/A, though small.

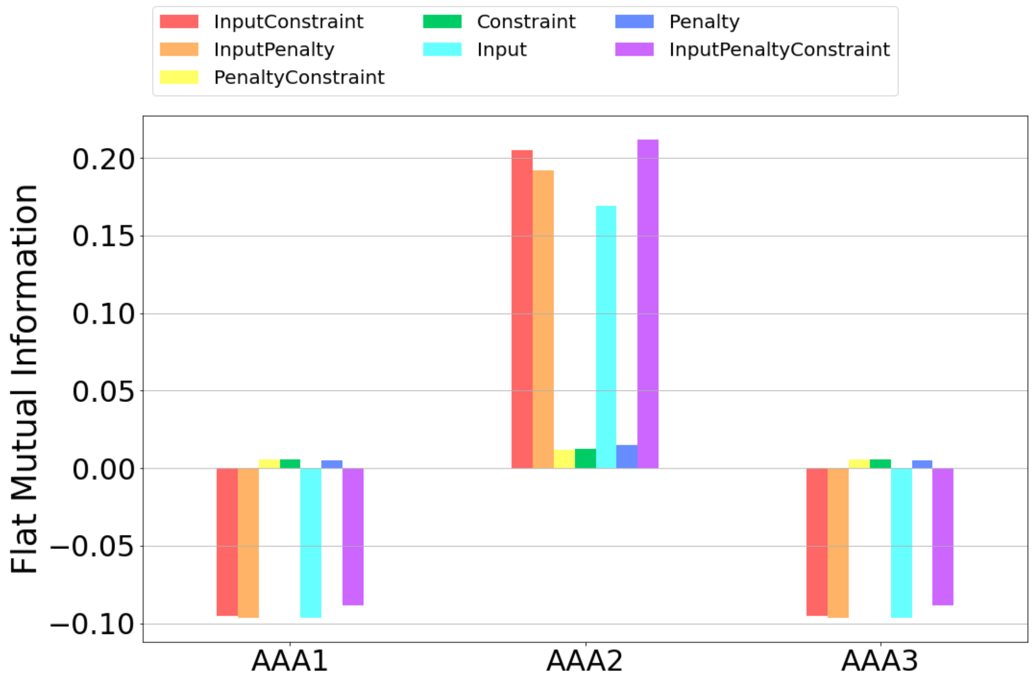

Figure 4.3: Flat Mutual Information: Case 2

	AAA1	AAA2	AAA3
Input Constraint	-0.096	0.205	-0.096
Input Penalty	-0.096	0.192	-0.096
Penalty Constraint	0.006	0.012	0.006
Constraint	0.006	0.013	0.006
Input	-0.097	0.169	-0.097
Penalty	0.005	0.015	0.005
Input Penalty Constraint	-0.088	0.212	-0.088

Table 4.31: Flat Mutual Information Values: Case 2

Flat Mutual Information

Figure 4.3 shows relatively large flat mutual information values (AAA1 and AAA3: -0.088 to -0.097, AAA2: 0.169 to 0.212) for truncation categories containing 'Input' across all three syllogisms when compared to truncation categories not containing 'Input'. In contrast to similar values from Case 1, these are small. All syllogisms have small values for truncation categories not containing 'Input' (between 0.005 and 0.015). Interestingly, AAA1 and AAA3 values for truncation containing 'Input' are negative, in a reversal of the expectation that only AAA2 would have negative flat mutual information values.

Generally, negativity here indicates some emergent relationship between and due to statements rather than an expected reduction in uncertainty, and while for abduction that may be acceptable and still result in sufficient decision-making, it is not likewise so for deduction and induction which by nature are more rigid reasoning instances. Thus, negative values here indicate lack of suitability of the generated data for knowledge queries.

The same differentiations that were present in Case 1 that imply the additive nature of truncation categories are present here in Case 2 as well. For AAA2, 'Input' is the smallest value and 'Input, Penalty, Constraint' is the largest with 'Input, Penalty' and 'Input, Constraint' in the middle. The same trend is present for AAA3 but the values are closer together as seen in Table 4.31.

It is apparent from the previously discussed probability distributions that truncations containing 'Input' are not likely to be sufficient for decision-making. Additionally, AAA1 and AAA3 are likely not sufficient either, due to the term assignment of this case. Possibly 'enough solutions' as the predicate term does not provide a strong enough descriptor about truncation in conclusion for decision-making.

Layered Mutual Information

	AAA1	AAA2	AAA3
Input Constraint	0.049	0.037	-0.096
Input Penalty	0.055	0.026	-0.098
Penalty Constraint	0.342	0.333	0.008
Constraint	0.344	0.328	0.01
Input	0.048	0.029	-0.088
Penalty	0.342	0.321	0.011
Input Penalty Constraint	0.046	0.023	-0.091

Table 4.32: Layered Mutual Information Values: Case 2

Figure 4.4 has an opposite trend than expected: truncation categories not con-

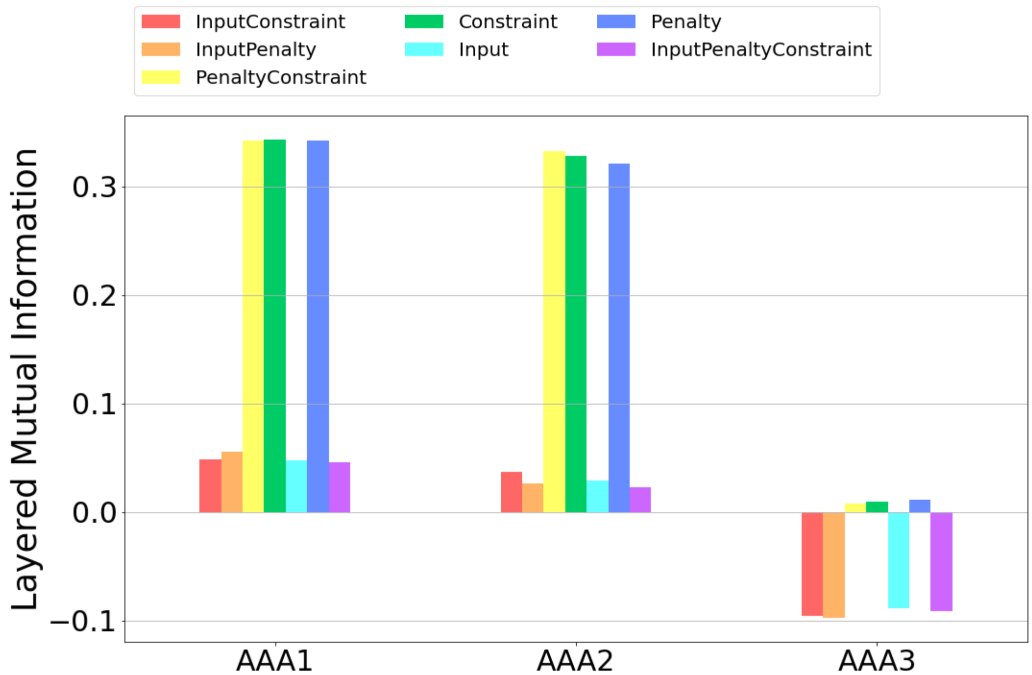

Figure 4.4: Layered Mutual Information: Case 2

taining 'Input' have larger layered mutual information values than those containing 'Input' for AAA1 and AAA2. The usual trend returns for AAA3. All values for layered mutual information are relatively small when compared with those seen in Case 1. Small layered mutual information values indicate a lack of suitability of generated data for knowledge queries.

The differentiation between truncation categories with 'Input' and those without is slightly less for AAA2 than AAA1, which emphasizes the insufficiency of AAA2 specifically. Abduction does not have the structure strength deduction has and so to also lack the connectivity between premises and conclusion implied by the layered mutual information values does not provide confidence of sufficiency in that decision instance.

Negative values for AAA3 indicate a large amount of non-relevant events in comparison to relevant events. Induction, which is the generalization of a trend from data, cannot be used appropriately in circumstances that lack data. This may change should term definitions be adjusted. These observations imply that Case 2 is not sufficient for decision-making.

4.4.4 Case 3

To enable the evaluation of generated tool data as design inferences for decision-making, probability distributions between statements and mutual information of the associated network for Case 3 are calculated. They are presented here.

CPTs

	Major = True		Major = False		Major = N/A	
Syllogism:	AAA1&AAA3	AAA2	AAA1&AAA3	AAA2	AAA1&AAA3	AAA2
Input Penalty Constraint	0.06	0.06	0.496	0.44	0.444	0.5
Input Constraint	0.069	0.069	0.5	0.431	0.431	0.5
Input	0.069	0.069	0.496	0.431	0.435	0.5
Input Penalty	0.075	0.075	0.496	0.425	0.429	0.5
Penalty	0.494	0.494	0.498	0.006	0.008	0.5
Penalty Constraint	0.494	0.494	0.498	0.006	0.008	0.5
Constraint	0.496	0.496	0.498	0.004	0.006	0.5

Table 4.33: Major Premise Probabilities: Case 3

Due to their syllogism construction, deduction and induction (AAA1 and AAA3) have the same major premise: ‘a useful optimizer can be a more truncated optimizer’. The major premise for abduction (AAA2) is ‘a more truncated optimizer is a useful optimizer’.

Table 4.33 shows differentiation in probability values for truncation categories containing ‘Input’. When the nodal status of the major premise is True, the differentiation occurs for all three syllogism constructions. When the nodal state of the major premise is False or N/A, the differentiation only appears according to ‘useful optimizer’ term positioning. For AAA1 and AAA3, ‘useful optimizer’ determines relevancy as the first term of the major premise and so the difference in truncation category probability values is visible for AAA1 and AAA3 when major premise nodal status is N/A. As the second term for AAA2, it determines if nodal status is True or False and can be seen when major premise nodal status is False.

	Minor = True		Minor = False		Minor = N/A	
Syllogism:	AAA1&AAA2	AAA3	AAA1&AAA2	AAA3	AAA1&AAA2	AAA3
Input Penalty Constraint	0.529	0.529	0.442	0.027	0.029	0.444
Input Constraint	0.531	0.531	0.429	0.038	0.04	0.431
Input	0.525	0.525	0.383	0.04	0.092	0.435
Input Penalty	0.54	0.54	0.423	0.031	0.038	0.429
Penalty	0.95	0.95	0.008	0.042	0.042	0.008
Penalty Constraint	0.965	0.965	0.008	0.027	0.027	0.008
Constraint	0.96	0.96	0.006	0.033	0.033	0.006

Table 4.34: Minor Statement Probabilities: Case 3

Deduction and abduction (AAA1 and AAA2) have the same minor premise due to their syllogism construction: ‘enough solutions indicate a useful optimizer’. The major premise for induction (AAA3) is ‘a useful optimizer produces enough solutions’.

Table 4.34 shows differentiation in probabilities for truncation categories containing ‘Input’. When the nodal status of the minor premise is True, the differentiation occurs for all three syllogism constructions. When the nodal state of the major premise is False or N/A, the differentiation only appears according to ‘useful optimizer’ term positioning. For AAA3, ‘useful optimizer’ determines relevancy as the first term of the minor premise and so differentiation in truncation category probability is visible for AAA3 when major premise nodal status is N/A. As the second term for AAA1 and AAA2, it determines if nodal state is True or False and can be seen when minor premise nodal status is False.

	Conclusion = True	Conclusion = False	Conclusion = N/A
Input Penalty Constraint	0.494	0.477	0.029
Input Constraint	0.496	0.465	0.04
Input	**0.442**	0.467	**0.092**
Input Penalty	0.488	0.475	0.038
Penalty	0.494	0.465	0.042
Penalty Constraint	0.498	0.475	0.027
Constraint	0.496	0.471	0.033

Table 4.35: Conclusion Statement Probabilities: Case 3

Table 4.35 has the same probability values for each nodal state for all syllogisms because all syllogisms in a case have the same conclusion: 'enough solutions are generated from more truncated optimizers'. Since 'useful optimizer' is the middle term for Case 3, there is no differentiation based off of truncation categories containing 'Input' for the conclusion statement as it is composed of the subject and predicate terms.

For all truncation categories, conclusion statements with a True nodal status are all greater than 0.44, with the lowest being the truncation category of 'Input' at 0.442 (bolded), about 0.05 less than most other truncation categories probability values. 'Input' truncation also has the largest of the small set of N/A conclusion status at 0.092. It is desirable to have a larger True conclusion statement nodal status as that indicates the definitions chosen to conduct this analysis with do not immediately preclude decision construction from the minor and major premises. For Case 3, all truncation categories have reasonably high conclusion statement nodal status True probabilities.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.931	0	0.069	0.954	0	0.046	0	0.995	0.005
Input Constraint	0.97	0	0.03	0.929	0	0.071	0	0.995	0.005
Input	0.909	0	0.091	0.933	0	0.067	0	0.88	0.12
Input Penalty	0.917	0	0.083	0.95	0	0.05	0	0.985	0.015
Penalty	0.987	0	0.013	0.929	0	0.071	0	1	0
Penalty Constraint	0.996	0	0.004	0.95	0	0.05	0	1	0
Constraint	0.992	0	0.008	0.941	0	0.059	0	1	0

Table 4.36: Minor Given Major Statement Probabilities for AAA1 Syllogism: Case 3

Table 4.36 shows the probabilities of the minor premise given the major premise for deduction (AAA1).

The zero probability values for False minor premise given True or False major premise and True minor premise given N/A major premise are due to the syllogism structure. The same is true for minor premise nodal status True given major premise nodal status N/A.

Table 4.37 shows the probabilities of the minor premise given the major premise

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.931	0	0.069	0	0.995	0.005	0.946	0.008	0.046
Input Constraint	0.97	0	0.03	0	0.995	0.005	0.929	0	0.071
Input	0.909	0	0.091	0	**0.879**	**0.121**	0.925	0.008	0.067
Input Penalty	0.917	0	0.083	0	0.985	0.015	0.942	0.008	0.05
Penalty	0.987	0	0.013	0	1	0	0.925	0.004	0.071
Penalty Constraint	0.996	0	0.004	0	1	0	0.946	0.004	0.05
Constraint	0.992	0	0.008	0	1	0	0.938	0.004	0.058

Table 4.37: Minor Given Major Statement Probabilities for AAA2 Syllogism: Case 3

for abduction (AAA2).

The probability values of zero for False minor premise given True major premise and True minor premise given False major premise are due to the syllogism structure. If the major premise has a True nodal status, then the minor premise nodal state must be either True or N/A (it cannot be False). If the major premise has a False nodal status, the minor premise nodal state must be False or N/A (it cannot be True). The structure of the syllogism requires this as AAA2 has the middle term as the second term in both the minor and major premises.

The bolded values are interesting. 'Input' has a slightly smaller False minor premise nodal status probability given False major premise nodal status, and also a corresponding relatively large N/A minor premise nodal status probability given False major premise nodal status when compared to other truncation categories that contain 'Input'.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.931	0.069	0	0.954	0.046	0	0	0	1
Input Constraint	0.97	0.03	0	0.929	0.071	0	0	0	1
Input	0.909	0.091	0	0.933	0.067	0	0	0	1
Input Penalty	0.917	0.083	0	0.95	0.05	0	0	0	1
Penalty	0.987	0.013	0	0.929	0.071	0	0	0	1
Penalty Constraint	0.996	0.004	0	0.95	0.05	0	0	0	1
Constraint	0.992	0.008	0	0.941	0.059	0	0	0	1

Table 4.38: Minor Given Major Statement Probabilities for AAA3 Syllogism: Case 3

Table 4.38 shows the probabilities of the minor premise given the major premise for induction (AAA3). The syllogism structure of induction has the middle term as the first term in both the minor and major premises, therefore making it so that the minor premise has a N/A nodal status if and only if the major premise nodal status is N/A.

	Conclusion |Major = True			Conclusion |Major = False			Conclusion |Major = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.931	0	0.069	0	0.954	0.046	0.986	0.009	0.005
Input Constraint	0.97	0	0.03	0	0.929	0.071	0.995	0	0.005
Input	0.909	0	0.091	0	0.933	0.067	**0.871**	0.01	**0.12**
Input Penalty	0.917	0	0.083	0	0.95	0.05	0.976	0.01	0.015
Penalty	0.987	0	0.013	0	0.929	0.071	0.75	0.25	0
Penalty Constraint	0.996	0	0.004	0	0.95	0.05	0.75	0.25	0
Constraint	0.992	0	0.008	0	0.941	0.059	0.667	0.333	0

Table 4.39: Conclusion Given Major Statement Probabilities for AAA1 and AAA3 Syllogisms: Case 3

Since deduction (AAA1) and induction (AAA3) have identical major premises and conclusions, Table 4.39 shows conclusion probabilities given major premise for both AAA1 and AAA3. Syllogism structure also requires that conclusion statements with nodal status False have zero probability given major premise status of True. Conclusion statements with True nodal status given major premise False nodal status also have zero probability.

Differentiation of truncation categories with and without 'Input' is evident for conclusion statement given major premise with True and N/A nodal status, and the differentiation is comparatively small for conclusion given major premise True.

Bolded values in Table 4.39 show that 'Input' has a smaller True conclusion nodal status probability given N/A major premise nodal status, and a corresponding relatively large N/A conclusion nodal status probability given N/A major premise nodal status when compared to other truncations including 'Input'. Interestingly, 'Input' truncation category probability is closer to the median probability values rather than the maximum.

Table 4.40 shows conclusion probabilities given major premise for abduction (AAA2). The expected differentiation along 'Input' truncation categories for conclusion state-

Conclusion Status:	Conclusion |Major = True True	False	N/A	Conclusion |Major = False True	False	N/A	Conclusion |Major = N/A True	False	N/A
Input Penalty Constraint	0.931	0	0.069	0.995	0	0.005	0	0.954	0.046
Input Constraint	0.97	0	0.03	0.995	0	0.005	0	0.929	0.071
Input	0.909	0	0.091	0.879	0	0.121	0	0.933	0.067
Input Penalty	0.917	0	0.083	0.985	0	0.015	0	0.95	0.05
Penalty	0.987	0	0.013	1	0	0	0	0.929	0.071
Penalty Constraint	0.996	0	0.004	1	0	0	0	0.95	0.05
Constraint	0.992	0	0.008	1	0	0	0	0.942	0.058

Table 4.40: Conclusion Given Major Statement Probabilities for AAA2 Syllogism: Case 3

ment given major premise True and False nodal states are very small. Conclusion status cannot be False if the major premise is relevant (does not have a N/A nodal status) and cannot be True if the major premise has a N/A nodal status. This is due to the AAA2 syllogism structure where the predicate term is the first term of the major premise and the second term of the conclusion.

Conclusion Status:	Conclusion |Minor = True True	False	N/A	Conclusion |Minor = False True	False	N/A	Conclusion |Minor = N/A True	False	N/A
Input Penalty Constraint	0.106	0.894	0	0.991	0.009	0	0	0	1
Input Constraint	0.125	0.875	0	1	0	0	0	0	1
Input	0.119	0.881	0	0.989	0.011	0	0	0	1
Input Penalty	0.127	0.873	0	0.99	0.01	0	0	0	1
Penalty	0.513	0.487	0	0.75	0.25	0	0	0	1
Penalty Constraint	0.51	0.49	0	0.75	0.25	0	0	0	1
Constraint	0.512	0.488	0	**0.667**	**0.333**	0	0	0	1

Table 4.41: Conclusion Given Minor Statement Probabilities for AAA1 and AAA2 Syllogisms: Case 3

Since deduction (AAA1) and abduction (AAA2) have identical minor premises and conclusions, Table 4.41 shows conclusion statement probabilities given minor premise for both AAA1 and AAA2. Syllogism structure also requires that conclusion with nodal status N/A have zero probability given minor premise states of True or False. AAA1 and AAA2 have the subject term as the first term in both the minor premise and the conclusion, which manifests as zero probabilities for conclusion nodal states True and False given minor premise N/A nodal status and a probability of one for conclusion N/A status given minor premise N/A status. The minor premise determines the relevance of the conclusion statement for deduction and abduction.

As expected, there is differentiation along 'Input' truncation categories for conclusion given minor premise True and False nodal states.

Also present is the previously discussed trend in regards to 'Constraint' truncation category True and False conclusion nodal state given False major premise status (bolded in Table 4.41).

	Conclusion |Minor = True			Conclusion |Minor = False			Conclusion |Minor = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0.894	0	0	0	1	0.986	0.009	0.005
Input Constraint	0.125	0.875	0	0	0	1	0.995	0	0.005
Input	0.119	0.881	0	0	0	1	0.871	0.01	0.12
Input Penalty	0.127	0.873	0	0	0	1	0.976	0.01	0.015
Penalty	0.513	0.487	0	0	0	1	0.75	0.25	0
Penalty Constraint	0.51	0.49	0	0	0	1	0.75	0.25	0
Constraint	0.512	0.488	0	0	0	1	**0.667**	**0.333**	0

Table 4.42: Conclusion Given Minor Statement Probabilities for AAA3 Syllogism: Case 3

Table 4.42 shows conclusion probabilities given minor premise for induction (AAA3). Syllogism structure of AAA3 requires that conclusion with nodal status N/A have zero probability given minor premise status of True. AAA3 has the subject term as the second term of the minor premise and the first term of the conclusion, which manifests as zero probabilities for conclusion nodal states True and False given minor premise False nodal status and a probability of one for conclusion N/A status given minor premise False status.

As expected, there is a differentiation between truncation categories including 'Input' for conclusion given minor status True and N/A.

Also present is the previously discussed trend in regards to 'Constraint' truncation category True and False conclusion nodal state given False major premise status (bolded).

Flat Mutual Information

Figure 4.5 and Table 4.43 show large flat mutual information values for truncation categories containing 'Input' (between 0.164 and 0.212). For those truncation

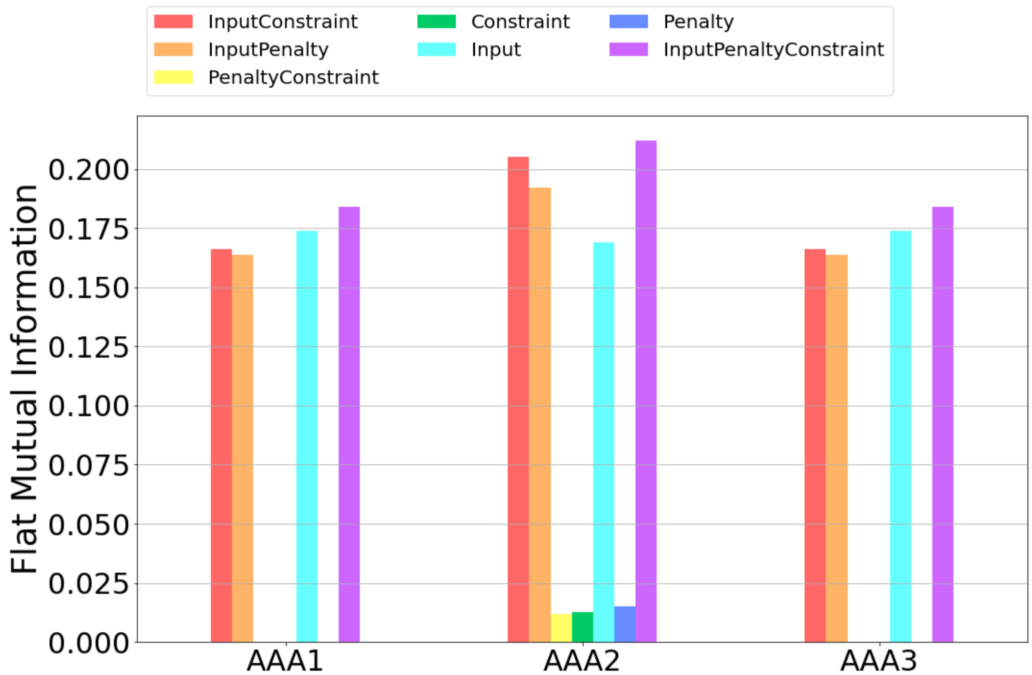

Figure 4.5: Flat Mutual Information: Case 3

	AAA1	AAA2	AAA3
Input Constraint	0.166	0.205	0.166
Input Penalty	0.164	0.192	0.164
Penalty Constraint	5E-5	0.012	5E-5
Constraint	1E-4	0.013	1E-4
Input	0.174	0.169	0.174
Penalty	1E-4	0.015	1E-4
Input Penalty Constraint	0.184	0.212	0.184

Table 4.43: Flat Mutual Information Values: Case 3

categories, AAA2 has the largest average values in comparison to AAA1 and AAA3 yet the lowest singular value for 'Input'. Unlike the additive nature expected of truncation categories, 'Input' has the second highest value for AAA1 and AAA3 after 'Input, Penalty, Constraint'. This deviation may imply that for this case structure in abductive circumstances, 'Input' is relatively more sufficient than other truncation categories in comparison to other reasoning instances.

Deduction and induction have flat mutual information values of zero for truncation categories without 'Input'. Abduction does not as its flat mutual information values are between 0.012 and 0.015 for those truncation categories. This implies some measure of dependence for AAA2 in comparison with the independent premises and conclusions of AAA1 and AAA3. Realistically, it can be interpreted as lack of evidence for these truncation categories, leaving it unable to be determined sufficient or not sufficient for decision-making though if no data exists for a circumstance than that circumstance cannot be sufficient for decision-making as there is simply nothing there to make a decision from.

It is apparent from the previously discussed probability distributions that truncation categories containing 'Input' are not likely to be sufficient for decision-making. Due to the magnitude of values for both AAA1/AAA3 and AAA2, truncation categories including 'Input' are inappropriate for sufficient decision-making.

Layered Mutual Information

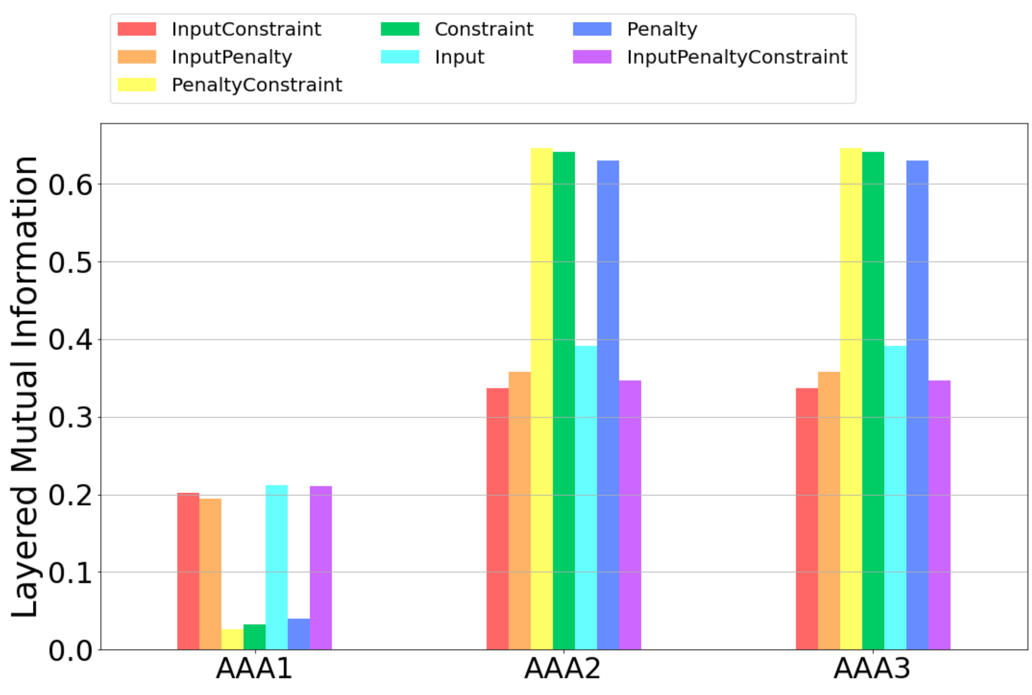

Figure 4.6: Layered Mutual Information: Case 3

Figure 4.6 and Table 4.44 show layered mutual information values for Case 3.

	AAA1	AAA2	AAA3
Input Constraint	0.202	0.337	0.336
Input Penalty	0.194	0.357	0.357
Penalty Constraint	0.026	0.646	0.646
Constraint	0.033	0.642	0.641
Input	0.212	0.391	0.391
Penalty	0.04	0.631	0.63
Input Penalty Constraint	0.21	0.347	0.347

Table 4.44: Layered Mutual Information Values: Case 3

Similar to Case 2, Case 3 has a trend for truncation categories containing 'Input' to be larger than those not containing 'Input'. For Case 3, those syllogisms in which this occurs are AAA2 and AAA3.

This implies a much stronger relationship between premises and conclusions for truncation categories not containing 'Input'. Possibly, it is also due to the middle term 'useful optimizer'. While large, these values are not as large as those seen in Case 1 implying that they may be more indicative of sufficient decision-making.

Despite being smaller than those truncation categories not containing 'Input', categories containing 'Input' for AAA2 and AAA3 are larger than those for AAA1. AAA1 values may be too low to for sufficient decision-making. Some concern is had for categories identified as likely lacking evidence as seen in the small AAA1 and large AAA3 layered mutual information values

4.4.5 Case 4

To enable the evaluation of generated tool data as design inferences for decision-making, probability distributions between statements and mutual information of the associated network for Case 4 are calculated. They are presented here.

	Major = True		Major = False		Major = N/A	
Syllogism:	AAA1&AAA3	AAA2	AAA1&AAA3	AAA2	AAA1&AAA3	AAA2
Input Penalty Constraint	0.494	0.494	0.477	0.006	0.029	0.5
Input Constraint	0.496	0.496	0.465	0.004	0.04	0.5
Input	**0.442**	**0.442**	0.467	0.058	**0.092**	0.5
Input Penalty	0.488	0.488	0.475	0.013	0.038	0.5
Penalty	0.494	0.494	0.465	0.006	0.042	0.5
Penalty Constraint	0.498	0.498	0.475	0.002	0.027	0.5
Constraint	0.496	0.496	0.471	0.004	0.033	0.5

Table 4.45: Major Premise Probabilities: Case 4

CPTs

Deduction and induction (AAA1 and AAA3) have the same major premise due to their syllogism construction: 'enough solutions are generated from more truncated optimizers'. The major premise for abduction (AAA2) is 'a more truncated optimizer produces enough solutions'.

Table 4.45 does not show differentiation in probabilities on truncation categories containing 'Input', due to the assignment of 'useful optimizer' as the subject term which is not present in the major premise. However, 'Input' truncation category's probability for all syllogisms when the major premise nodal status is True is 0.442, which is noticeably lower than other truncation category probabilities for the same premise status. 'Input' truncation category for AAA1 and AAA3 when major premise nodal status is N/A is 0.092, which is larger than other truncation category probabilities.

Since the predicate term is 'more truncation' and determines relevancy for AAA2, conclusion N/A nodal status probabilities for AAA2 are 0.5 due to the method of generating evidence (where even amounts of 1, 2, 3, and 4-digit truncation occurred for all dataset runs).

Deduction and abduction (AAA1 and AAA2) have the same minor premise due to their syllogism construction: 'a useful optimizer produces enough solutions'. The major premise for induction (AAA3) is 'enough solutions indicate a useful optimizer'.

Table 4.46 shows separation in probabilities values for truncation categories with

	Minor = True		Minor = False		Minor = N/A	
Syllogism:	AAA1&AAA2	AAA3	AAA1&AAA2	AAA3	AAA1&AAA2	AAA3
Input Penalty Constraint	0.529	0.529	0.027	0.442	0.444	0.029
Input Constraint	0.531	0.531	0.038	0.429	0.431	0.04
Input	0.525	0.525	0.04	**0.383**	0.435	0.092
Input Penalty	0.54	0.54	0.031	0.423	0.429	0.038
Penalty	0.95	0.95	0.042	0.008	0.008	0.042
Penalty Constraint	0.965	0.965	0.027	0.008	0.008	0.027
Constraint	0.96	0.96	0.033	0.006	0.006	0.033

Table 4.46: Minor Statement Probabilities: Case 4

and without 'Input'. When the nodal status of the minor premise is True, the differentiation occurs for all three syllogism constructions. When the nodal state of the major premise is False or N/A, the differentiation only appears according to 'useful optimizer' term positioning. For AAA1 and AAA2, 'useful optimizer' determines relevancy as the first term of the minor premise and so differentiation in truncation category probability is visible for AAA1 and AAA2 when major premise nodal status is N/A. As the second term for AAA3, it determines if the nodal state is True or False and can be seen when minor premise nodal status is False.

Minor premise with False nodal status for truncation categories containing 'Input' are all greater than 0.38, with the lowest being the truncation category of 'Input' at 0.383 (bolded), about 0.04 less than the next lowest truncation category containing 'Input'.

	Conclusion = True	Conclusion = False	Conclusion = N/A
Input Penalty Constraint	0.06	0.496	0.444
Input Constraint	0.069	0.5	0.431
Input	0.069	0.496	0.435
Input Penalty	0.075	0.496	0.429
Penalty	0.494	0.498	0.008
Penalty Constraint	0.494	0.498	0.008
Constraint	0.496	0.498	0.006

Table 4.47: Conclusion Statement Probabilities: Case 4

Table 4.47 has the same values for each nodal state for all syllogisms because all syllogisms in a case have the same conclusion: 'a useful optimizer can be a more truncated optimizer'. Since 'useful optimizer' is the subject term for Case 4, the differentiation based off of truncation categories containing 'Input' for the conclusion statement is for nodal states True and N/A as the subject term determines relevance.

	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
Minor Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.114	0	0.886	0.991	0	0.009	0	0.929	0.071
Input Constraint	0.134	0	0.866	1	0	0	0	0.947	0.053
Input	0.142	0	0.858	0.991	0	0.009	0	**0.432**	**0.568**
Input Penalty	0.141	0	0.859	0.991	0	0.009	0	0.833	0.167
Penalty	0.987	0	0.013	0.996	0	0.004	0	1	0
Penalty Constraint	0.987	0	0.013	0.996	0	0.004	0	1	0
Constraint	0.992	0	0.008	0.996	0	0.004	0	1	0

Table 4.48: Minor Given Major Statement Probabilities for AAA1 Syllogism: Case 4

Table 4.48 shows the probabilities of the minor premise given the major premise of deduction (AAA1). As expected, differentiation on truncation categories containing 'Input' occurs for minor premise given major premise nodal states of True and N/A, and not for False major premise nodal status, the same way that Table 4.45 has truncation differentiation. This differentiation is much more pronounced for minor premise given major premise True than for minor premise given major premise N/A.

The probability values of zero for False minor premise given True or False major premise and True minor premise given N/A major premise are due to the syllogism structure.

There is a large difference in probability values between 'Input' and any other truncation category for minor premise given major premise nodal status N/A. 'Input' truncation's probability value for minor premise False nodal status given major premise N/A nodal status is 0.432, approximately half the next smallest probability value for another truncation category. This is an indication that something odd is happening in regards to 'Input' truncation category evidence for major premise nodal status N/A, likely the same thing as what causes the AAA1 and AAA3 probability value of 0.092 for major premise N/A nodal status seen in Table 4.45.

Table 4.49 shows the probabilities of the minor premise given the major premise for abduction (AAA2). As expected, differentiation for truncation categories containing

Minor Status:	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.114	0	0.886	0	0.667	0.333	0.946	0.046	0.008
Input Constraint	0.134	0	0.866	0	0.5	0.5	0.929	0.071	0
Input	0.142	0	0.858	0	0.107	0.893	0.925	0.067	0.008
Input Penalty	0.141	0	0.859	0	0.5	0.5	0.942	0.05	0.008
Penalty	0.987	0	0.013	0	1	0	0.925	0.071	0.004
Penalty Constraint	0.987	0	0.013	0	1	0	0.946	0.05	0.004
Constraint	0.992	0	0.008	0	1	0	0.938	0.058	0.004

Table 4.49: Minor Given Major Statement Probabilities for AAA2 Syllogism: Case 4

'Input' occurs for minor premise given major premise nodal states of True and False. This differentiation is much more pronounced for minor premise given major premise True than for minor premise given major premise N/A.

The probability values of zero for False minor premise given True major premise and True minor premise given False major premise are due to the syllogism structure. If the major premise has a True nodal status, then the minor premise nodal state must be either True or N/A (it cannot be False). If the major premise has a False nodal status, the minor premise nodal state must be False or N/A (it cannot be True). The structure of the syllogism requires this as AAA2 has the middle term as the second term in both the minor and major premises.

Minor Status:	Minor |Major = True			Minor |Major = False			Minor |Major = N/A		
	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.114	0.886	0	0.991	0.009	0	0	0	1
Input Constraint	0.134	0.866	0	1	0	0	0	0	1
Input	0.142	0.858	0	0.991	0.009	0	0	0	1
Input Penalty	0.141	0.859	0	0.991	0.009	0	0	0	1
Penalty	0.987	0.013	0	0.996	0.004	0	0	0	1
Penalty Constraint	0.987	0.013	0	0.996	0.004	0	0	0	1
Constraint	0.992	0.008	0	0.996	0.004	0	0	0	1

Table 4.50: Minor Given Major Statement Probabilities for AAA3 Syllogism: Case 4

Table 4.50 shows the probabilities of the minor premise given the major premise for induction (AAA3). As expected, differentiation for truncation categories containing 'Input' occurs for minor premise given major premise nodal status of True.

The syllogism structure of induction assigns the middle term as the first term in both the minor and major premises, therefore making it so that the minor premise has a N/A nodal status if and only if the major premise nodal status is N/A.

	Conclusion |Major = True			Conclusion |Major = False			Conclusion |Major = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.114	0	0.886	0	0.991	0.009	0.143	0.786	0.071
Input Constraint	0.134	0	0.866	0	1	0	0.053	0.895	0.053
Input	0.142	0	0.858	0	0.991	0.009	0.068	**0.364**	**0.568**
Input Penalty	0.141	0	0.859	0	0.991	0.009	0.167	0.667	0.167
Penalty	0.987	0	0.013	0	0.996	0.004	0.15	0.85	0
Penalty Constraint	0.987	0	0.013	0	0.996	0.004	0.077	0.923	0
Constraint	0.992	0	0.008	0	0.996	0.004	0.125	0.875	0

Table 4.51: Conclusion Given Major Statement Probabilities for AAA1 and AAA3 Syllogisms: Case 4

Deduction (AAA1) and induction (AAA3) have identical major premises and conclusions, thus Table 4.51 presents probabilities for conclusion statements given major premise for both AAA1 and AAA3. Syllogism structure requires that conclusion statements with nodal status False have zero probability given major premise status of True. Conclusion statements with True nodal status for major premise False nodal status also have zero probability.

The bolded values in Table 4.51 are interesting. 'Input' has a much smaller False conclusion statement nodal state probability given N/A major premise nodal status, and also a corresponding relatively large N/A conclusion statement nodal state probability given N/A major premise nodal status when compared to other truncation categories including 'Input'. For truncation categories without 'Input', when the conclusion statement nodal status is N/A given a N/A major premise nodal status, the probability is zero. Truncation categories containing 'Input' for the same circumstances have small probabilities, which implies some not yet understood effect such as lack of evidence.

Table 4.52 shows the probabilities of the minor premise given the major premise for induction (AAA3). As expected, differentiation for truncation categories with and without 'Input' occurs for minor premise given major premise nodal states of True and False.

Conclusion Status:	Conclusion |Major = True True	False	N/A	Conclusion |Major = False True	False	N/A	Conclusion |Major = N/A True	False	N/A
Input Penalty Constraint	0.114	0	0.886	0.667	0	0.333	0	0.992	0.008
Input Constraint	0.134	0	0.866	0.5	0	0.5	0	1	0
Input	0.142	0	0.858	**0.107**	0	**0.893**	0	0.992	0.008
Input Penalty	0.141	0	0.859	0.5	0	0.5	0	0.992	0.008
Penalty	0.987	0	0.013	1	0	0	0	0.996	0.004
Penalty Constraint	0.987	0	0.013	1	0	0	0	0.996	0.004
Constraint	0.992	0	0.008	1	0	0	0	0.996	0.004

Table 4.52: Conclusion Given Major Statement Probabilities for AAA2 Syllogism: Case 4

The syllogism structure of induction has the middle term as the first term in both the minor and major premises, therefore making it so that the minor premise has a N/A nodal status if and only if the major premise nodal status is N/A.

'Input' truncation category has a much smaller True conclusion statement nodal state probability given False major premise nodal status, and also a corresponding relatively large N/A conclusion statement nodal state probability given False major premise nodal status when compared to all other truncation categories, most notably the other categories that contain 'Input'. For truncations without 'Input', when the conclusion statement nodal status is True or N/A given a False major premise nodal status, the probability is one or zero respectively. This is in contrast with truncation categories containing 'Input' for the same circumstances that have approximate probability values of 0.5.

Conclusion Status:	Conclusion |Minor = True True	False	N/A	Conclusion |Minor = False True	False	N/A	Conclusion |Minor = N/A True	False	N/A
Input Penalty Constraint	0.106	0.894	0	0.154	0.846	0	0	0	1
Input Constraint	0.125	0.875	0	0.056	0.944	0	0	0	1
Input	0.119	0.881	0	0.158	0.842	0	0	0	1
Input Penalty	0.127	0.873	0	0.2	0.8	0	0	0	1
Penalty	0.513	0.487	0	0.15	0.85	0	0	0	1
Penalty Constraint	0.51	0.49	0	0.077	0.923	0	0	0	1
Constraint	0.512	0.488	0	0.125	0.875	0	0	0	1

Table 4.53: Conclusion Given Minor Statement Probabilities for AAA1 and AAA2 Syllogisms: Case 4

Since deduction (AAA1) and abduction (AAA2) have identical minor premises and conclusions, Table 4.53 shows conclusion probabilities given minor premise for both AAA1 and AAA2. Syllogism structure also requires that conclusion with nodal status N/A have zero probability given minor premise states of True or False. AAA1 and AAA2 have the subject term as the first term in both the minor premise and the conclusion, which manifests as zero probabilities for conclusion nodal states True and False given minor premise N/A nodal status and a probability of one for conclusion N/A status given minor premise N/A status. The minor premise determines the relevance of the conclusion statement for deduction and abduction.

As expected, there is differentiation between truncation categories containing 'Input' and those not for conclusion statements given a True minor premise nodal status.

	Conclusion |Minor = True			Conclusion |Minor = False			Conclusion |Minor = N/A		
Conclusion Status:	True	False	N/A	True	False	N/A	True	False	N/A
Input Penalty Constraint	0.106	0.894	0	0	0	1	0.143	0.786	0.071
Input Constraint	0.125	0.875	0	0	0	1	0.053	0.895	0.053
Input	0.119	0.881	0	0	0	1	0.068	0.364	0.568
Input Penalty	0.127	0.873	0	0	0	1	0.167	0.667	0.167
Penalty	0.513	0.487	0	0	0	1	0.15	0.85	0
Penalty Constraint	0.51	0.49	0	0	0	1	0.077	0.923	0
Constraint	0.512	0.488	0	0	0	1	0.125	0.875	0

Table 4.54: Conclusion Given Minor Statement Probabilities for AAA3 Syllogism: Case 4

Table 4.54 shows probability values for conclusion statements given minor premise for induction (AAA3). Syllogism structure of AAA3 requires that conclusion with nodal status N/A has zero probability given minor premise status of True. AAA3 has the subject term as the second term of the minor premise and the first term of the conclusion, which manifests as zero probability for conclusion nodal states True and False given minor premise False nodal status and a probability of one for conclusion N/A status given minor premise False status.

As expected, there is a difference between values for truncation categories with and without 'Input' for conclusion statements given minor status True.

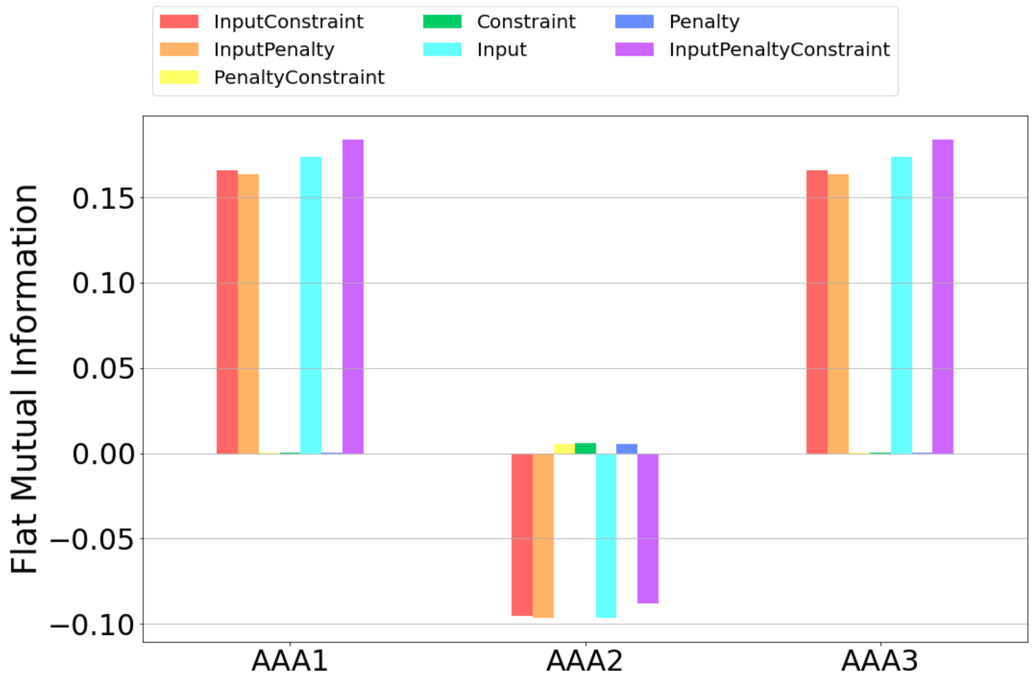

Figure 4.7: Flat Mutual Information: Case 4

	AAA1	AAA2	AAA3
Input Constraint	0.166	-0.096	0.166
Input Penalty	0.164	-0.096	0.164
Penalty Constraint	0	0.006	0
Constraint	0	0.006	0
Input	0.174	-0.097	0.174
Penalty	0	0.005	0
Input Penalty Constraint	0.184	-0.088	0.184

Table 4.55: Flat Mutual Information Values: Case 4

Flat Mutual Information

Figure 4.7 and Table 4.55 present flat mutual information values for Case 4. There is a clear distinction between truncation categories with and without 'Input'. AAA1 and AAA3 have values between 0.164 and 0.184 for truncation categories containing Input. AAA2 has values between -0.088 and -0.097 for the same categories.

Abduction having negative values does not indicate a lack of sufficiency to make decisions. Abduction, in comparison to deduction and induction, possesses a less rigorous structure connecting statements. Abduction articulates an insight or hunch that can derive from statements with weaker relationships. It is possible that the negative value indicates an emergent relationship between premises and conclusions rather than an increase in uncertainty.

Truncation categories not containing 'Input' are non-zero only for AAA2, where they range in mutual information values between 0.005 and 0.006. This implies a slight measure of dependence for AAA2 in comparison with the independent premises and conclusions of AAA1 and AAA3. Realistically, it can be interpreted as lack of evidence for these truncation categories, leaving it unable to be determined sufficient or not sufficient for decision-making. This lack of evidence is similar to Case 3.

The overall shape and trends of Figure 4.7 is reminiscent of Case 1. Both Case 4 and Case 1 possess 'enough solutions' as a middle term. This exemplifies the middle term's ability to connect the premises and thus provide a foundation upon which the conclusion is based.

Layered Mutual Information

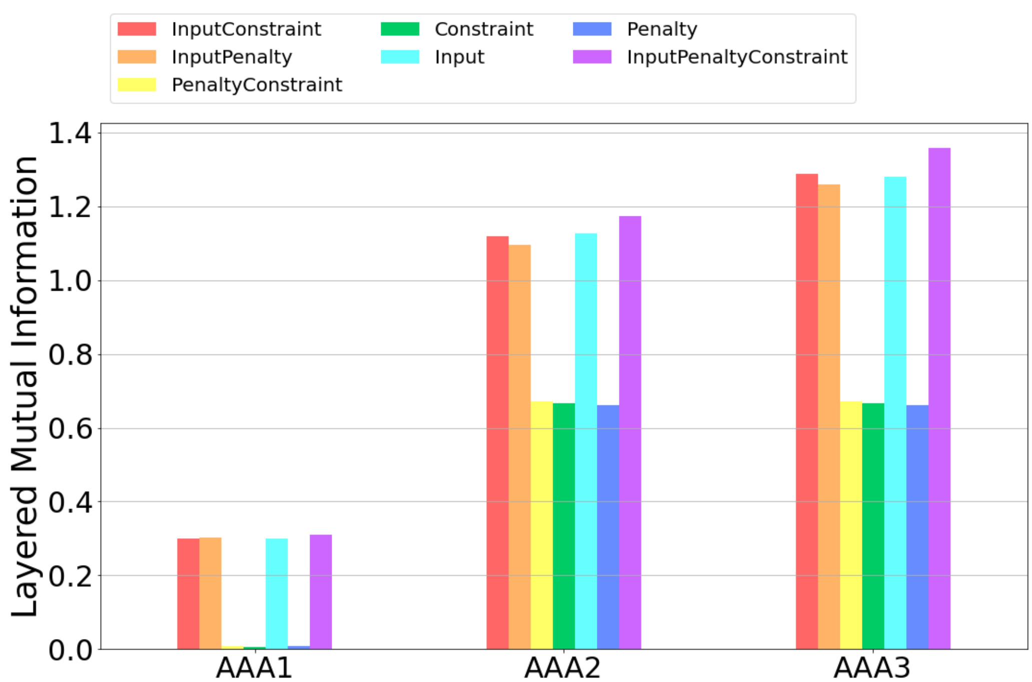

Figure 4.8: Layered Mutual Information: Case 4

	AAA1	AAA2	AAA3
Input Constraint	0.3	1.118	1.289
Input Penalty	0.303	1.096	1.259
Penalty Constraint	0.008	0.672	0.672
Constraint	0.006	0.668	0.668
Input	0.3	1.126	1.279
Penalty	0.008	0.662	0.662
Input Penalty Constraint	0.31	1.173	1.359

Table 4.56: Layered Mutual Information Values: Case 4

Figure 4.8 and Table 4.56 present the layered mutual information values for Case 4. All three syllogisms show distinct differences in values for truncation categories with and without 'Input'. Truncation categories without 'Input' are very small (0.006 to 0.008) for AAA1 and much larger for AAA2 and AAA3 (0.662 to 0.672)

The rigidity of deductive syllogism constructions and the near-zero layered mutual information values indicate a lack of sufficiency for decision-making. Abduction and induction truncation categories have larger values and thus suggest sufficiency.

4.5 Sufficiency

To evaluate tool appropriateness, conclusion sufficiency for decision-making must first be discussed.

Probability values for all cases show that premise definitions strongly impact the correlation of conclusion values to major and minor premises. Low values, those close to zero, indicate a need to moderately revise definitions for the cases presented in order to ensue there is more data to draw conclusions from. Case 1 and Case 2 both strongly correlate the cognizance of a True minor premise to a True conclusion for truncation categories not containing 'Input'. Case 3 and Case 4 both strongly correlate the cognizance of a True major premise to a True conclusion for truncation categories not containing 'Input'. For all cases, truncation categories with 'Input' do not stand out.

Flat mutual information values can be separated into three groups across cases. The structure of the flat network used to calculate these values predicates certain syllogisms with specific statements to have identical trends. This is because flat mutual information does not differentiate between premises and conclusions and so lacks the ordering implications that are indicative of design decisions. Each flat mutual information value is essentially predicated on the repeated second term of the statements though still allowing for insights into the impact of terms within premises. Recall that Case 1 and Case 2 have the same subject term: 'more truncation'. Case 1 abductive syllogism, Case 2 deductive and inductive syllogisms, and Case 4 abductive syllogisms have identical flat mutual information values where truncation categories with 'Input' are negative and truncation categories without 'Input' are positive and small. These identical values are because of the term 'enough solutions', which especially requires a revised definition.

Case 3 deductive and inductive syllogisms and Case 4 deductive and inductive syllogisms have identical flat mutual information trend values as well. Truncation categories without 'Input' have zero mutual information, in this case an indication of lack of evidence and not independence. This is because of the term definitions of the first terms in the statement pair that predicates flat mutual information for these situations: 'enough solutions' and 'useful optimizer'. Case 1 deductive and inductive syllogisms, Case 2 abductive syllogism, and Case 3 abductive syllogisms have the largest flat mutual information values despite separation between truncation categories that contain 'Input' and those that do not, due to the relative irrelevance of 'enough solutions' not being a predicating term.

Due to the limitations of the flat representation, a layered representation was created. Mutual information from the layered network is calculated and utilized to contextually examine premises and conclusion relationships. A clear preference of deduction for decision-making is seen when all case syllogisms are compared. Case

1 is the most sufficient for decision-making, in part due to the sensible semantic implications in comparison to other cases. Assignment of terms for Case 1 indicated that the primary concern, 'truncation', should be the subject. The middle term should be defined such that there is appropriate amounts of evidence to connect the premises and conclusion statements. Thus, Case 1 shows an inclination for truncation categories not including 'Input' to be sufficient for decision-making, specifically in a deductive reasoning instance. More investigation is required to increase confidence to be sure that data generated and constructed using deductive inferences of Case 1 is suitable for knowledge queries.

Case 2 is not sufficient for decision-making. Negative layered mutual information values raise serious concerns about Case 2's term assignment. Case 2 lacks the necessary relationships between premises and statements needed in reasoning instances, likely due to the positioning of 'enough solutions' as the predicate term which precludes a strong connection between premises. In order to continue investigating sufficient decision-making, additional case constructions must be examined.

Case 3 is more suitable than Case 2. However, the startling flip of values of truncation categories containing 'Input' and those not containing 'Input' does not match what the dataset validation suggests: that datasets for truncations containing 'Input' are not sufficient for decision-making. It seems likely that due to the conclusion statement "enough solutions are generated from more truncated optimizers", Case 3 is borderline for strong enough inferences for sufficient decision-making in inductive reasoning instances but lacks the strength of correlation needed for deductive reasoning instances.

Case 4 has too large layered mutual information values to draw sufficient conclusions from inductive or abductive syllogisms and too small values for deductive syllogisms. The connection between premises allows for more suitable decisions to be made.

Of the cases considered, overall deductive syllogisms are the best chance for sufficiency. Deduction is a more rigorously structured syllogism than induction or abduction, and that is why in circumstances with such related terms as the ones expressed in these cases it is the closest to sufficiency. In order to make decisions from this tool, revised terms should be constructed from the designer's new hunch to alter the hypothesis for further investigation.

Compared premises suggest the importance of the middle term in establishing context between the major and minor premises. Without relevant data connecting the foundations of a conclusion, it is exceedingly difficult to generate inferences useful for decision-making. Designer construction and ordering of statements in syllogisms matter in addition to designer-defined terms.

4.6 Conclusions

Evaluating tool appropriateness requires understanding of conclusion sufficiency. Designer perspective is encoded within every step of design, and this is evident within this framework as the declaration and definition of terms.

Often, the designer can only identify something as *not* sufficient, and continue from there. Such is the situation with these cases where truncation categories that include 'Input' are not sufficient for decision-making. These truncation categories have been evaluated to find the tool not appropriate when truncated in such a way. This aligns with understanding that the consequences of truncating 'Input' are greater than that of 'Penalty' or 'Constraint' because of the compounding and progression of error. 'Input' values are the baseline of all optimizer calculations while 'Penalty' and 'Constraint' are more meta-evaluative as they compute a measure of unfitness and weigh the fitness score, so truncation of those categories do not have the same long reaching effect

This case study demonstrated the value of the framework proposed in Chapter

III. Syllogisms with different term assignments were evaluated for sufficiency. Probability, flat mutual information, and layered mutual information values were evaluated as quantifiable multi-level measures for making sufficient decisions. Subject, predicate, and middle terms were compared for impact within statements. Statement ordering was evaluated to distinguish syllogism constructions with layered mutual information.

CHAPTER V

Contributions

5.1 Contributions

This dissertation has presented a unique framework to answer the question, "how can designers determine the sufficiency or quality of a design decision knowledge query based upon design tool-generated data?"

The novel contributions of this research include:

- Realization that sufficiency is not just statistical success or an artifact of the tool (such as convergence statistics or Pareto optimality) but rather based upon actual and true logic theory grounded comparative evaluations associated with designer decision-making knowledge query utilizing available tool-generated data.

- Developed the novel ability to comparatively evaluate derived design data utilizing distinct syllogism constructs.

- Created a mechanism to model various decision-making knowledge queries in terms of deduction, abduction, and induction. These logical reasoning instances are described in logical syllogistic forms as AAA1, AAA2, and AAA3 respectively. A novel contribution of this thesis is to be able to evaluate each distinctly in context of the logical inference unique to that reasoning instance.

  - This methodology is the first in the marine domain and to the best of the author's knowledge, the first in the engineering design domain to do so.

- Identified required syllogistic factors of the major premise, minor premise, and conclusion statements to extend to multi-level representation.

- Development of multi-level network (flat and layered) of deductive (AAA1), abductive (AAA2), and inductive (AAA3) syllogisms, including the mechanisms for combining major and minor premises as compound constructions for layered networks, in order to evaluate both isolated and combined premise and conclusion relationships, enabling differentiation between major and minor premise contributions to conclusion statements.

- Developed unique metrics for enabling the evaluation of sufficiency with the application of probability and mutual information from information theory.

- Recognition of the need for probability statistics to evaluate sufficiency associated with logical syllogisms. A unique Bayesian framework was created to determine the probabilistic associations within and between the major, minor, and conclusion statements to evaluate sufficiency judgements based upon the ensemble design data.

  - Uniquely generate the probabilities necessary for eventual multi-level evaluation for sufficiency determination.

  - Determination of proper ordering of statements relative to context dependent on the conclusion to determine relevancy of associated tool-generated data.

  - Evaluation of sequence of premise statements to determine required directionality of statements for conclusion.

- Developed four instances of a primary case study that:

  - Validated and verified the framework.
  - Demonstrated the use of the flat and layered network representations in order to quantify the separate and combined impact of major and minor premises on conclusion constructions and thus interpretation of tool data suitability.
  - Demonstrated the need for layered networks to quantify the separate and combined impact of major and minor premises on conclusion constructions and thus interpretation of tool data suitability.
  - Demonstrated the impact of ordering of major, minor and conclusion statements in terms of syllogistic sufficiency.
  - Demonstrated the distinction between deduction, induction, and abduction as knowledge queries.

5.2 Future Topics of Interest

The focus of this dissertation was to assist in designer determination of the sufficiency of a design decision based upon design tool-generated data. With the completion of this work there are several future topics that are worth investigating. These future topics include, but are not limited to, the following:

- *Modification of network states from pseudo-Boolean logic to a multi-state environment.* Network states presented in this work were used to calculate the metrics for evaluation of tool use suitability. These states consisted of True, False, and N/A. While the addition of N/A allowed for discernment of non-relevant data and provided proof of concept for the framework, further sophistication could lead to more nuanced transformations of semantic language to quantified statements. Fuzzy logic may be one way in which to do this.

- *Extension to other syllogism forms beyond AAA1, AAA2, and AAA3.*
  - Syllogisms with figure four structures are cyclical and thus unable to be represented as a Bayesian network without some sort of term transformation. AAA4 specifically is another form of abduction that may provide additional insights as the relationships between statements are differently correlated than the AAA1, AAA2, and AAA3 syllogisms examined in this work.
  - Additionally, this work only evaluated syllogisms without negatory premises or conclusions. It can be useful in design to determine when tool data is suitable to draw conclusions about the lack of something. Syllogism mood types E or O can be investigated for these purposes.
  - Comparison of individual syllogisms with premise moods of differing strength of conviction. For example, syllogisms of structures AII and IAI under the current framework would result in the same evaluation. Adjustment of semantic transformations to differentiate 'some' and 'all' as language modifiers could provide additional granularity in designer intent and suitable use of tool-generated data.
- *Application of the framework to larger, more complex decisions.* The layered network methodology provides the basic technique to combine multiple statements. Additional tiers for this would allow more sophisticated insights on ordering implications for decision-making.
- *Inclusion of other network metrics to evaluate more nuanced insights of decisions.* This work currently uses probability distributions and mutual information. With more complex combinations of premises, additional statistical analyses or metrics could quantify and isolate impacts of specific statement contributions to decision-making. Additionally, with increased combinations

of statements, concepts such as Markov Blankets may be useful to isolate the impact of grouped statements to decision-making, or as contributions to a sub-decision.

- Articulation and development of additional methods to quantify the impact of cognitive biases on tool use suitability distinct from designer subjectivity.

APPENDIX

APPENDIX A

Bulk Carrier Synthesis Model

This appendix provides an overview of the bulk carrier synthesis model developed by *Sen and Yang* (1998). Additional information may also be found in *Yang et al.* (1990) and *Yang and Sen* (1996).

Inputs and Intermediate Functions

The model defines six inputs: length (L), beam (B), draft (T), depth (D), speed (V), and block coefficient (C_B). These inputs can be expressed by the input vector $\boldsymbol{x} = [L, B, T, D, V, C_B]$. Using these inputs, the model defines a host of intermediate functions:

$$\begin{aligned} \text{annual cost} &= \text{capital charges} + \text{running cost} \\ &\qquad + \text{voyage cost} + \text{RTPA} \end{aligned} \tag{A.1}$$

$$\text{capital charges} = 0.2 \times \text{ship cost} \tag{A.2}$$

$$\begin{aligned} \text{ship cost} &= 1.3 \times (\text{steel mass})^{0.85} \\ &\qquad + 3500 \times \text{outfit mass} + 2400 \times P^{0.8} \end{aligned} \tag{A.3}$$

$$\text{steel mass} = 0.034 \times L^{1.7} \times B^{0.7} \times D^{0.4} \times C_B^{0.5} \tag{A.4}$$

$$\text{outfit mass} = L^{0.8} \times B^{0.6} \times D^{0.3} \times C_B^{0.1} \tag{A.5}$$

$$\text{machinery mass} = 0.17 \times P^{0.9} \tag{A.6}$$

$$\text{P} = \Delta^{2/3} \times V^3 \times \frac{1}{b(c_B) \times \frac{V}{(g \times L)^{0.5}} + a\,(C_B)} \tag{A.7}$$

$$\Delta = 1.025 \times L \times B \times T \times C_B \tag{A.8}$$

$$\text{running cost} = 40000 \times DW^{0.3} \tag{A.9}$$

$$DW = \Delta - \text{light ship mass} \tag{A.10}$$

$$\text{voyage cost} = \text{fuel cost} + \text{port cost} \tag{A.11}$$

$$\text{fuel cost} = 1.05 \times \text{daily consumption} \times \text{sea days} \times \text{fuel price} \tag{A.12}$$

$$\text{daily consumption} = P \times 0.19 \times 0.024 + 0.2 \tag{A.13}$$

$$\text{sea days} = \frac{\text{round trip miles}}{24 \times V} \tag{A.14}$$

$$\text{round trip miles} = 5000\,(\text{nautical miles}) \tag{A.15}$$

$$\text{fuel price} = 100\,(\text{pounds/ton}) \tag{A.16}$$

$$\text{port cost} = 6.3 \times DW^{0.8} \tag{A.17}$$

$$\text{RTPA} = \frac{350}{\text{sea days} + \text{port days}} \tag{A.18}$$

$$\text{port days} = 2 \times \left(\frac{\text{cargo deadweight}}{\text{cargo handling rate}} + 0.5\right) \tag{A.19}$$

$$\text{cargo deadweight} = DW - \text{fuel carried} - \text{crew, stores, and water} \tag{A.20}$$

$$\text{fuel carried} = \text{daily consumption} \times (\text{sea days} + 5) \tag{A.21}$$

$$\text{crew, stores, and water} = 2.0 \times DW^{0.5} \tag{A.22}$$

$$\text{cargo handling rate} = 8000\,(\text{tons/day}) \tag{A.23}$$

where $RTPA$ is round trips per annum, DW is deadweight, and g is the gravitational constant ($g = 9.8065\ m/s^2$). The functions $a(C_B)$ and $b(C_B)$ are regression equations based on Froude Number and a coefficient referred to as the *Admiralty Coefficient*, detailed in the original paper.

Objectives

The model defines three objectives:

$$\Omega_1 = \min(\text{transportation cost}) \tag{A.24}$$

$$\Omega_2 = \min(\text{light ship mass}) \tag{A.25}$$

$$\Omega_3 = \max(\text{annual cargo}) \tag{A.26}$$

which can be expressed by the objective vector $\mathbf{\Omega} = [\Omega_1, \Omega_2, \Omega_3]$. The functions that comprise the individual objectives are defined in terms of the model's intermediate functions as:

$$\text{transportation cost} = \frac{\text{annual cost}}{\text{annual cargo}} \tag{A.27}$$

$$\text{light ship mass} = \text{steel mass} + \text{outfit mass} + \text{machinery mass} \tag{A.28}$$

$$\text{annual cargo} = \text{cargo deadweight} \times \text{RTPA} \tag{A.29}$$

Constraints

The model defines dimensional and displacement constraints:

$$L/B \geq 6 \tag{A.30}$$

$$L/D \leq 15 \tag{A.31}$$

$$L/T \leq 19 \tag{A.32}$$

$$T \leq 0.45 \times DW^{0.31} \tag{A.33}$$

$$T \leq 0.7 \times D + 0.7 \tag{A.34}$$

$$DW \geq 3000 \tag{A.35}$$

$$DW \leq 500000 \tag{A.36}$$

powering constraints:

$$C_B \geq 0.63 \tag{A.37}$$

$$C_B \leq 0.75 \tag{A.38}$$

$$V \geq 14 \tag{A.39}$$

$$V \leq 18 \tag{A.40}$$

$$\frac{V}{(g \times L)^{0.5}} \leq 0.32 \tag{A.41}$$

and a stability constraint:

$$GM \geq 0.07 \times B \tag{A.42}$$

where

$$GM = KB + BM - KG \tag{A.43}$$

$$KB = 0.53 \times T \tag{A.44}$$

$$BM = \frac{(0.085 \times C_B - 0.002) \times B^2}{T \times C_B} \tag{A.45}$$

$$KG = 1.0 + 0.52 \times D \tag{A.46}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

Ammon, S. (2017), Why designing is not experimenting: Design methods, epistemic praxis and strategies of knowledge acquisition in architecture, *Philosophy & Technology*, *30*, 495–520.

Andrews, D. J. (2012), Art and science in the design of physically large and complex systems, *Proceedings of the Royal Society*, *A*, 891–912, doi:10.1098/rspa.2011.0590.

Brefort, D. (2018), Managing Epistemic Uncertainty in Design Models through Type-2 Fuzzy Logic Multidisciplinary Optimization, Ph.D. thesis, University of Michigan.

Bullock, J. B., and S. H. Logan (1969), A model for decision making under uncertainty, *Agricultural Economics Research*, *21*(4), 109–115.

Conklin, J. (2006), *Dialogue Mapping: Building Shared Understanding of Wicked Problems*, John Wiley & Sons, West Sussex, England.

Dong, A. (2016), Functional lock-in and the problem of design transformation, *Research in Engineering Design*, *28*, 203–221.

Dong, X., and C. C. Hayes (2011), Using uncertainty to inform information sufficiency in decision making, in *Engineering Psychology and Cognitive Ergonomics*, edited by D. Harris, pp. 294–302, Springer Berlin Heidelberg, Berlin, Heidelberg.

Edwards, S., M. Cilli, T. Peterson, M. Zabat, C. Lawton, and L. Shelton (2015), Whole Systems Trade Analysis, in *24th Annual INCOSE International Symposium*, vol. 24.

Einhorn, H. J., and R. M. Hogarth (1981), Behavioral decision theory: Processes of judgment and choice, *Journal of Accounting Research*, *19*(1), 1–31.

Einhorn, H. J., and R. M. Hogarth (1987), Decision making: going forward in reverse, *Harvard Business Review*, (January-February), 66–70.

Ferguson, E. S. (1994), Engineering and the mind's eye.

Floridi, L. (2017), The logic of design as a conceptual logic of information, *Minds and Machines*, *27*, 495–519.

Flyvbjerg, B., M. Garbuio, and D. Lovallo (2009), Delusion and Deception in Large Infrastructure Projects: Two Models for Explaining and Preventing Executive Disaster, *California Management Review*, *51*(2), 170–193.

Goodrum, C. J. (2020), Conceptually Robust Knowledge Generation in Early Stage Complex Design, Ph.D. thesis, University of Michigan.

Hanson, N. R. (1958), The logic of discovery, *The Journal of Philosophy*, *55*(25), 1073–1089.

Hattori, M. (2016), Probabilistic representation in syllogistic reasoning: A theory to integrate mental models and heuristics, *Cognition*, *157*, 296 – 320.

Henry, S. M., L. A. Waddell, and M. R. DiNunzio (2016), The Whole Systems Trades Analysis Tool for Autonomous Ground Systems, in *NDIA Ground Vehicle Systems Engineering and Technology Symposium*.

Howard, R. (1977), Decision analysis in systems engineering, in *Readings in Decision Analysis*, pp. 47–81, Stanford Research Institute.

Jensen, F. V., and T. D. Nielsen (2007), *Bayesian Networks and Decision Graphs*, Springer, New York, NY.

Kassel, B., S. Cooper, and A. Mackenna (2010), Rebuilding the NAVSEA Early Stage Ship Design Environment, *Tech. rep.*, American Society of Naval Engineers (ASNE), Arlington, VA.

Knight, J. T., and D. J. Singer (2014), Applying real options analysis to naval ship design, *2014 ASNE Conference Proceedings*.

Lazar, N. (1938), The importance of certain concepts and laws of logic for the study and teaching of geometry, *The Mathematics Teacher*, *31*(4), 156–174.

NAVSEA (2012), Ship Design Manager (SDM) and Systems Integration Manager (SIM) Manual, *Tech. rep.*

Newman, M. E. J. (2010), *Networks: An Introduction*, Oxford University Press, Oxford.

Page, S. E. (2006), Path dependence, *Quarterly Journal of Political Science*, *1*, 87–115.

Parsons, M., H. Chung, E. Kirtley, A. Daniels, S. Liu, and J. Patel (2008), Intelligent ship arrangements: A new approach to general arrangement, *Naval Engineers Journal*, *120*, 51 – 65, doi:10.1111/j.1559-3584.2008.00153.x.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Peirce, C. S. (1967), *Annotated Catalogue of the Papers of Charles S. Peirce and The Peirce Papers: A supplementary catalogue*, University of Massachusetts Press.

Rollier, B., and J. A. Turner (1994), Planning forward by looking backward: Retrospective thinking in strategic decision making, *Decision Sciences*, *25*(2), 169–188.

Schneidman, E., S. Still, M. J. Berry, and W. Bialek (2003), Network information and connected correlations, *Physical Review Letters*, *91*(23).

Sen, P., and J.-B. Yang (1998), Multiple Objective Decision Making, in *Multiple Criteria Decision Support in Engineering Design*, chap. 4.4.1, pp. 150–157, Springer.

Shepherd, D. A., and K. M. Sutcliffe (2011), Inductive top-down theorizing: A source of new theories of organization, *The Academy of Management Review*, *36*(2), 361–380.

Shields, C. P. F. (2017), Investigating Emergent Design Failures Using a Knowledge-Action-Decision Framework, Ph.D. thesis, University of Michigan.

Singer, D. J., N. Doerry, and M. E. Buckley (2009), What Is Set-Based Design?, *Naval Engineers Journal*, *121*(4), 31–43.

Sober, E. (2013), *Core Questions in Philosophy: A Text with Readings*, 6 ed., Boston: Pearson Education.

Spero, E., C. L. Bloebaum, B. J. German, A. Pyster, and A. M. Ross (2014), A Research Agenda for Tradespace Exploration and Analysis of Engineered Resilient Systems, in *Conference on Systems Engineering Research*.

Sypniewski, M. J. (2019), A Novel Analysis Framework for Evaluating Predisposition of Design Solutions through the Creation of Hereditary-Amelioration Networks Derived from the Dynamics within an Evolutionary Optimizer, Phd thesis, University of Michigan.

Taylor, R., N. A. Torugsa, and A. Arundel (2018), Leaping into real-world relevance: An "abduction" process for nonprofit research, *Nonprofit and Voluntary Sector Quarterly*, *47*(1), 206–227.

Tessler, M., and N. Goodman (2014), Some arguments are probably valid: Syllogistic reasoning as communication, *Proceedings of the Annual Meeting of the Cognitive Science Society*, *36*.

Thompson, B. (2016), Deductively valid, inductively valid, and retroductively valid syllogisms, *Transactions of the Charles S. Peirce Society*, *52*(4), 611–632.

Tsujishita, T. (1995), On triple mutual information, *Advances in Applied Mathematics*, *16*(3), 269–274.

Webber, M. M., and H. W. J. Rittel (1973), Dilemmas in a General Theory of Planning, *Policy Sciences*, *4*(December 1969), 155–169.

Yang, J.-B., and P. Sen (1996), Interactive trade-off analysis and preference modeling for preliminary multiobjective ship design, *Systems Analysis, Modelling and Simulation*, *26*, 25–55.

Yang, J.-B., C. Chen, and Z. J. Zhang (1990), The Interactive Step Trade-Off Method (ISTM) for Multiobjective Optimization, *IEEE Transactions on Systems, Man and Cybernetics*, *20*(3), 688–695.